A DEMONSTATION OF THE METHOD OF STOCHASTIC FINITE ELEMENT ANALYSIS

THESIS

Paul R. Bryant
Captain, USAF

AFIT/GA/AA/89M-01

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GA/AA/89M-01

A DEMONSTATION OF THE METHOD OF STOCHASTIC FINITE ELEMENT ANALYSIS

THESIS

Paul R. Bryant
Captain, USAF

AFIT/GA/AA/89M-01

AFIT/GA/AA/89M-01

A DEMONSTATION OF THE METHOD OF STOCHASTIC FINITE ELEMENT ANALYSIS

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Intitute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Astronautical Engineering

Paul R. Bryant, B.S.
Captain, USAF

March, 1989

Approved for public release; distribution unlimited

Acknowledgements

Though my name appears as the author of this thesis, there are many others whose contributions constitute the very underpinnings of this final product. As my faculty advisor, Major David Robinson (Dr. Dave) kept me moving in the right direction with "what ifs" and "what nexts". I am grateful to him for his patience and sponsorship; I think we both learned more than we thought we would. Many thanks to my thesis commettee members, Major Ron Heinrichsen of AFWAL, and Captain Howard Gans. Their insight helped fill in the empty spots. Thanks to Captain Phil Beran; as system manager of the VAX computer, he made sure sufficient resources were always available to me. Bob Trame of Information Systems really helped in getting ASTROS up and running.

Of course, neither this thesis nor my entire tour at AFIT, would have been possible, or even bearable, were it not for my family. My lovely wife Robyn kept the household running and the home fires burning the months I was on remote assignment in the basement, and my three young'uns, Jill, Jake and little Chelsea Elizabeth finally get their dad back--I missed you too, kids.

Table of Contents

List of Figures

Figure

Abstract

Finite element analysis has been used as a design tool for many years, with structural reliability being ensured through use of a liberal factor of safety. Unfortunately, the safety factor is a blanket insurance against all hazards, and a designer has no way to optimize a structure against any particular hazard. This is particularly troublesome in the fields of aero/astro design, where every bit of mass must serve to maximum utility.

The method of Stochastic Finite Element Analysis allows a designer to model any loading or hazard condition as closely to reality as desired by using an appropriate probability distribution function. Through a Monte Carlo simulation, the finite element model is subjected to the probability functions. The cumulative output is analyzed for trends in failure probability and the design is altered to enhance its reliability, repeating the process until the desired level of reliability is achieved. The resulting design is optimal for the imposed conditions, and compared to a structure designed with a traditional factor of safety approach, is either lighter or more reliable. This demonstration revealed that for similar reliabilities, a stochastically designed wing was 20% lighter than a wing strengthened by the factor of safety.

The major drawback in applying the method of stochastic finite element analysis is that very large, complex models can require extraordinary amounts of computer resources.

A DEMONSTRATION OF THE METHOD

STOCHASTIC FINITE ELEMENT ANALYSIS

Introduction

Traditional structural design has evolved constantly over the years. For example, early cathedrals were designed much stronger than required for structural soundness because their designers didn't understand structural mechanics and used extreme overdesign to avoid any possibility of structural collapse. (5:259) As engineers came to understand the mechanics of structures, they were able to reduce the size of supporting structure without reducing its ability to safely carry a load. Nevertheless, because of uncertainties in material properties and loading conditions, as well as inexact methods of modeling reality, structures were always designed to be stronger than needed. This extra capacity, or Factor of Safety (FS) is routinely applied in structural design as insurance against failure due to both predictable and unpredictable loads and structural characteristics. Nearly every civil work constructed in the United States is built according to a legislated safety factor, as dictated by such codes as the Uniform Building Code (7:455) and the Manual of Steel Construction (2:Sec 2,21).

Use of the safety factor has worked well primarily because it is simple to apply and it works well in providing a sound,

stable structure. It provides a reasonable means to account for the perhaps dozens of variables that might affect the ability of any given structure to withstand a required load. The designer has only to reduce the maximum allowable stress of the structural material by the factor of safety and proceed with the design and analysis.

Is there any reason to abandon this simple, reliable means of ensuring satisfactory performance? If weight is a concern, then the answer is yes. In the quest for higher performance aircraft and greater useful orbital payloads, every pound within the load bearing structure should ideally be stressed to just short of failure. Any lightly stressed structural element contains excess structural weight and does not represent optimal design.

Obviously, through careful design and the use of a factor of safety of one, this result can be obtained. A problem occurs, however, when there are undetected structural defects. Perhaps, the material is just a little weaker than planned for, or the load profile might be slightly more active than anticipated. A design engineer should be able to compensate for these potential failures, however rare they might be, without resorting to the broad brush approach of the factor of safety? Additionally, suppose the designer assumes that some part of the structure contains a flaw or that a localized structural failure has occured and wants to

assess its affect on the rest of the structure. Obviously, then, an engineer must account for these hidden weaknesses, design uncertainties, and assumed damages with an optimal design and still obtain a robust, durable structure.

If he can estimate what is the chance of the occurrence of any given event, then he can plan his design accordingly. If he knows that only one weld of every ten thousand is defective, and the structure has only ten welds, he can design appropriately. Concurrently, if he knows that the occurence of even one bad weld will precipitate complete structural collapse and loss of life, he can adjust the design to compensate for that too. It is this situation which brings us to the foundation of this thesis.

A body of statistical mathematics along with standard finite element analysis techniques have been combined into a method called stochastic finite element analysis. The method is completely computer-oriented and relies on the computer's ability to do lengthy finite element calculations quickly and repetitively. The recent advent of cheap, high-speed computer time makes this technique feasible because of the method's potential for using large amounts of computer resources. With traditional finite element analysis, a structure only had to be analyzed once. With stochastic finite element analysis, a structure might have to be analyzed hundreds, even thousands of times.

Problem Statement

Stochastic Finite Element Analysis (SFEA) is a method of finite element analysis wherein the ultimate product of the analysis is not the computed stress a given load creates within a structure, but rather, a probability that the structure will successfully withstand a given load.

Finite Element Analysis (FEA) is a tool. Its purpose is to simulate real world conditions, utilizing a mathematical model, bolstered up by underlying assumptions about that model. For example, we can generate a finite element mesh representing a wing. The model consists of various types of quad elements, shear panels and connecting rods to model the skin surfaces, ribs and spars of the simulated wing. Then, for purposes of the analysis, we assume the wings material properties are constant and that the loading to which it is subjected represents reality.

These models may be very good, perhaps the result of years of data collection and study. However, when the model and its assumptions are coded into an input deck for the finite element analysis, only one set of operational conditions is represented, not the continuum of changing conditions present in reality. Moreover, because input values may be averages

or modified by factors of safety, the simulation of reality may be inaccurate.

Nevertheless, the results of the finite element method are generally acceptable. Buildings don't fall down often and airplanes operate for long periods without their wings tearing off. However, when reality has not been modeled closely enough, or when the answers are not precise enough, the need to depart from traditional FEA emerges. Traditional finite element analysis allows neither the model not its assumptions to change. SFEA allows the designer to capture the previously discussed modeling and structural uncertainties in a manner more closely approximating reality. For example, the designer knows the wind loading on a wing is not static but varies by small gusts and buffets. Perhaps tests have disclosed that a certain percentage of connections will be flawed or there is a chance certain structural elements may be weakened or destroyed by external forces during operation. These can now be accounted for.

In SFEA, the design engineer allows these parameters to vary according to known (or assumed) probability distribution functions. However, because the internal workings of finite element programs require the input to be composed of deterministic functions, we are led to the fundamental difference

between implementing stochastic and traditional finite element models.

Traditionally, a finite element analysis is run only once, the output of stresses and strains are checked, and if needed, adjustments are made to the model or its assumptions. The process may be repeated to check the adjustments, but typically, only a few iterations are performed. In SFEA, the process is repeated a large number of times, utilizing Monte Carlo simulation (3:274) to model the uncertainties in material and structural properties. Within each simulation run, the analysis is deterministic; that is, it is performed exactly like any other finite element analysis. Once each run is completed, the results are stored away in a cumulative data file until enough data has been collected to allow statistically valid calculations. The output of these standard statistical calculations is a probability that an element or assembly of elements will not exceed some previously defined criterion.

Methods and Results

The finite element model used in this demonstration of the method of SFEA was a model of the Intermediate Complexity Wing, (see Figure 1) designed and provided by the Structures Division, Wright Research and Development Center, Wright-Patterson Air Force Base, OH. It depicts the wing of a small, modern, high performance fighter aircraft, somewhat resembling an F-5. The loading is aerodynamic and simulates supersonic flight. The load was scaled to bring the most highly stressed elements right to the edge of maximum allowable stress. This effectively removed any excessive design by reducing the factor of safety to 1.0. This hypothetical wing's purpose is to provide engineers with a reliable, benchmark model to help in formulating techniques and theories and for use in comparisons. Its intermediate complexity refers to the size of the elemental mesh. It is fine enough to yield good analysis results, but is coarse enough to be reasonably economical in terms of computer time.

In this demonstration of technique, the analysis was oriented toward evaluating the survivability of the wing, assuming it has suffered the loss of one structural element. This represents the wing suffering damage from a non-specific external source. This could mean a bird strike, being hit by a large caliber projectile, such as a bullet, or any other type of damage which would destroy the structural integrity of the element during flight. Although the model consists of 158

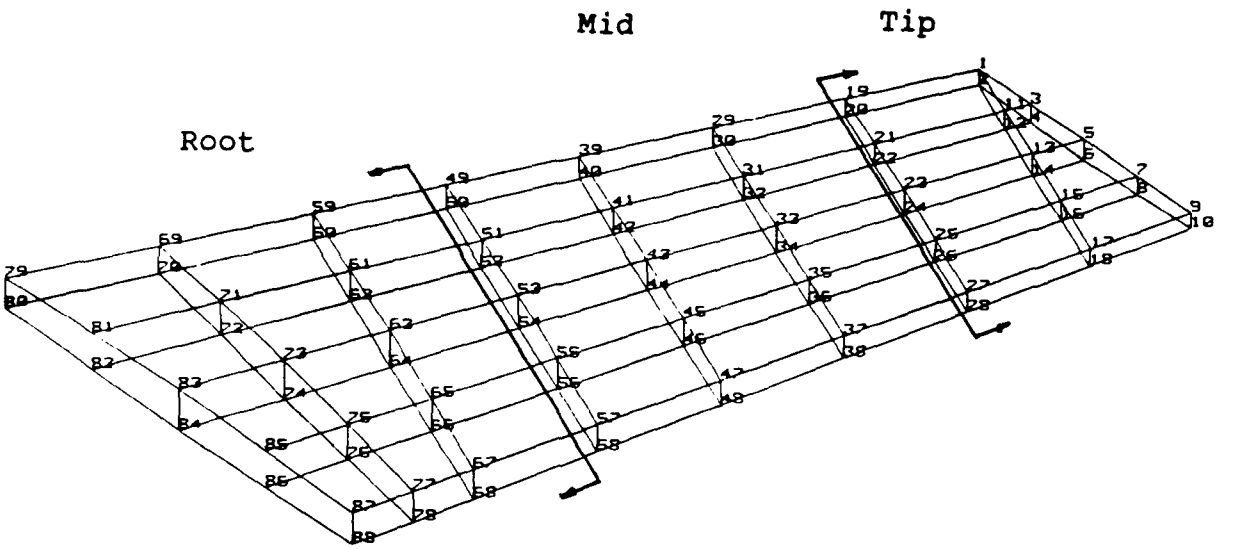


Figure 1. Intermediate Complexity Wing

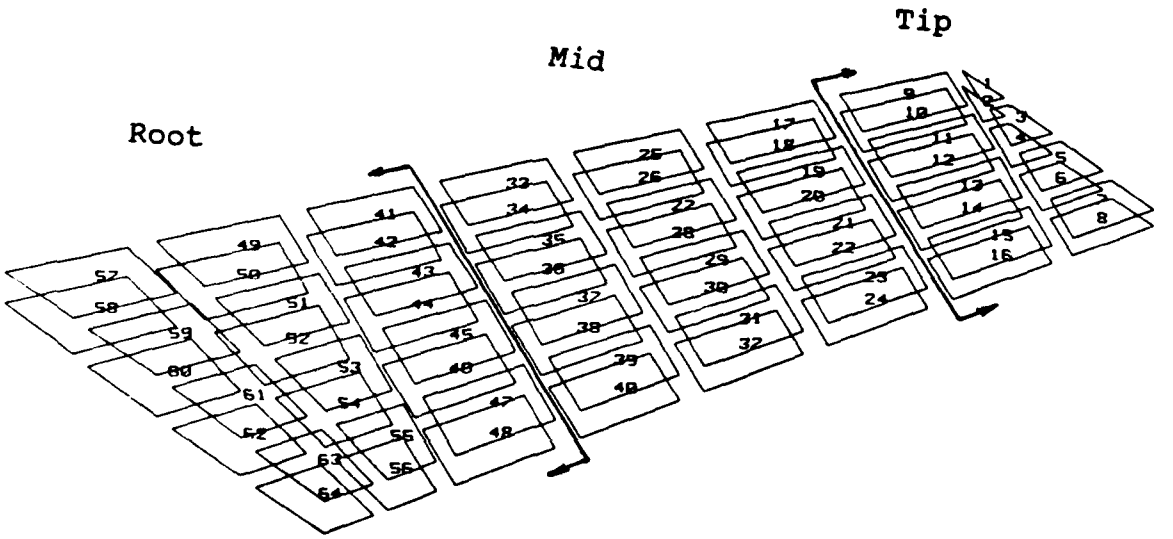


FIGURE 2. QUAD PANELS OF WING

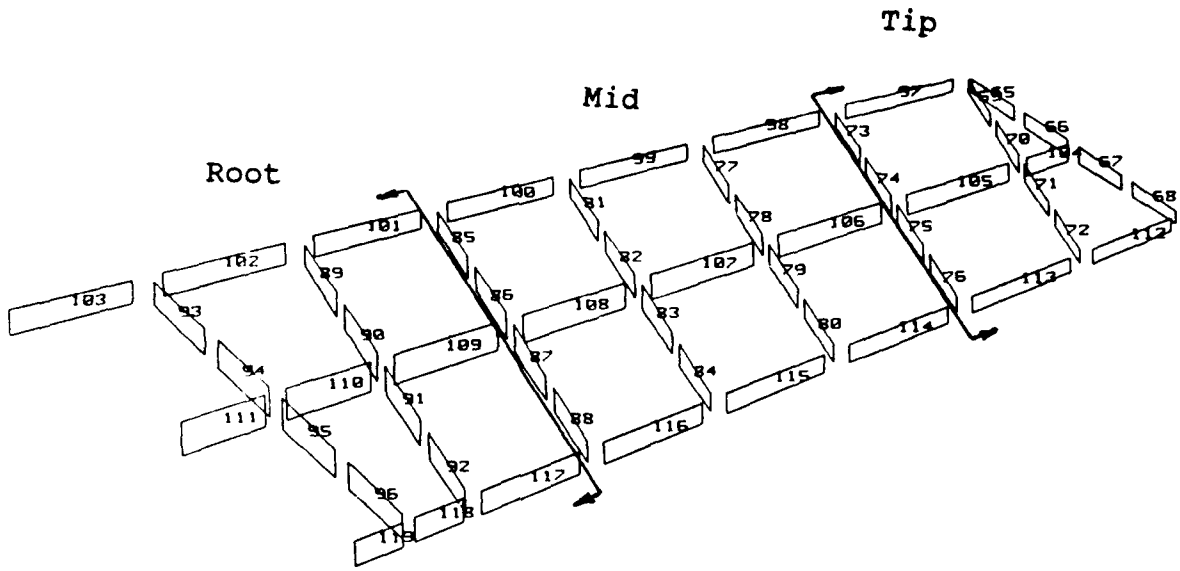


Figure 3. Shear Panels of Wing

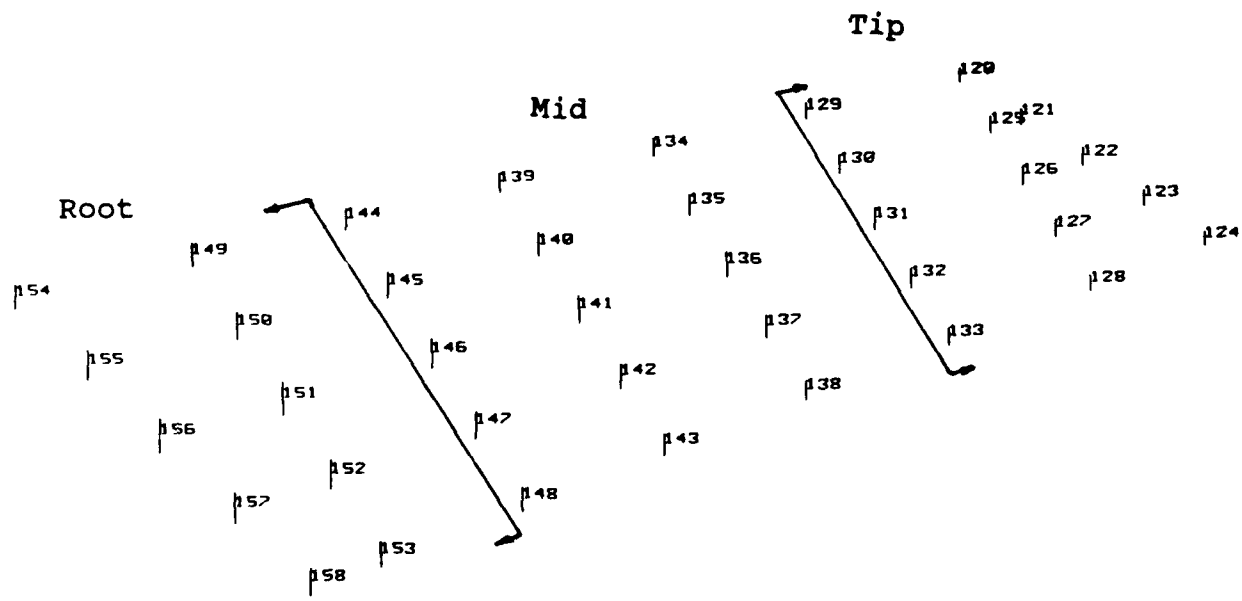


FIGURE 4. ROD ELEMENTS OF WING

elements, only 119 were considered susceptible to damag (55 shear panels, (Figure 2), 2 triangular elements and 62 quad elements, (Figure 3)). This is because 39 of the elements are rod elements (Figure 4) whose function is to hold the upper and lower wing surfaces apart. In reality, ribs and spars would perform this function, but in the model, these ribs and spars are represented by shear panels incapable of resisting axial stress. Thus, the model contains members not present in reality, but needed to make the model computationally correct.

For this demonstration, two quantities were selected as random variables. The first random variable was the structural element to be damaged. Damage was simulated by a reduction in the modulus of elasticity of the randomly selected element by a factor of 100; it could no longer carry any significant stress. The other random variable was the aerodynamic loading at each node. This load was assumed to vary according to a Gaussian, or normal, distribution.

At the beginning of each analysis, a FORTRAN coded preprocessor, called PRE.FOR, was run to assemble, from an unaltered, original data deck, the input data deck needed for the Monte Carlo simulation. The preprocessor generated a random number corresponding to an element to be damaged. As an additional evaluation tool, the wing was divided into approximate thirds: root, mid, and tip (see Figures 1, 2, 3, and 4 for divisions.) The wing taken as a whole, that is, damage not restricted to any particular area, is referred to

as the combined wing. The purpose of dividing the wing into sections was to allow insight into possible interrelations between damage in one element and stresses in another, perhaps some distance away. As it turned out, (and is discussed later), there indeed was a significant interelement stress relation. For example, damage in the tip section resulted in stress failures in even the root elements.

For the first one third of the runs, damage was allowed to occur in the root section only. If the randomly selected element was not within the root section, it was ignored and another was picked. Because all elements in a finite element model are uniquely numbered, this discrimination process was simple. Essentially, a uniform random number generator (8:195-199) produced a random integer between 1 and 119 (corresponding to the number of damageable elements.) This number was compared to a list of elements contained within the root section. Likewise, for the second series of runs, only the elements within the mid section were allowed to be damaged. The final third of the runs had damage restricted to the tip section. (See Appendix D for more detail about the selection process.)

The likelihood that any one element type was selected for damage was based on the surface that particular element type presented normal to the surface of the wing. The surface area of the wing in the "z" or "up" direction was about 1925 square units. The area of the front of the wing (the leading edge

shear panels) was about 180 square units. The surface area of the shear elements in the "z" direction was about 5 square units. Adding these areas together, then taking the approximate fraction of the shear panel area to the total area yielded the approximate ratio of one to ten. Thus, large, flat panels, the quad elements, were about ten times more likely to be "hit" than the thin ribs and spars, (i.e., the shear elements). Additionally, the quad and triangular elements were always damaged in pairs--the top panel and the panel immediately below it--simulating the way a projectile might pass through both. This partitioning of damage neglects several damage modes deemed unnecessarily complex for this demonstration. For example, a projectile might penetrate the wing at an oblique angle, damaging a lower quad panel, one or more shear panels, and possibly one or more additional quad panels on its way out. Or, a bird strike to the leading edge could easily damage several leading edge shear and quad elements. Nevertheless, algorithms could be written to accomodate these effects. Frequency and effects of bird-strikes are well studied and the effects of oblique angle projectile damage could be coded to the level of detail required.

Once a random element within the proper section was selected, the preprocessor read through the original data deck looking for the selected element's data cards. Once found, the preprocessor generated new cards giving the affected element, or elements, reduced load carrying capacity by reducing the

modulus of elasticity. Next, the nodal force cards were read in. The original forces for each node were considered the mean for that node and the preprocessor varied the forces about that mean with an associated coefficient of variation of 0.1 (9). Finally, the preprocessor completed the new data deck by copying the rest of the original data deck, such as grid coordinates and constraints, to the end of the new deck.

Next, this input data deck was fed into the finite element analysis program called ASTROS (Automated STRuctural Optimization System.) Again, this was supplied by the Structures Division of WRDC. ASTROS was instructed to perform stress and displacement calculations for all elements within the model. Ultimately, the displacement data was discarded as not being useful within the scope of this demonstration. Because of a lack of meaningful displacement-oriented failure criteria (other than that which could be obtained through stress-strain calculations), any failure mode based on excessive displacement would have been arbitrary and meaningless.

At the end of the analysis, another FORTRAN program, called POST1.FOR (for postprocessor, level 1) searched the ASTROS output data file. It looked for elemental stresses, combining them according to the Von Mises-Hencky Stress Criterion (10:85)

$$\text{stress} = s = (x^2 + y^2 - xy)^{1/2} \quad (1)$$

where:

x = major principal stress
y = minor principal stress

Maximum allowable stress was computed this same way using 55 ksi allowable tensile stress and 45 ksi allowable compressive stress, for a maximum allowable stress of 50.745 ksi (6). These elemental stresses were written to the end of a cumulative raw elemental stress data file for further analysis. The entire process was further automated by invoking a simple batch file to run sequentially PRE.FOR, ASTROS, and POST1.FOR a specified number of times, with an average run time for the sequence of three programs of about 65 cpu seconds (150 seconds, real time.) These programs were run on a Digital Equipment Corporation VAXstation II GPX minicomputer, operating at 3 mips.

For baseline calculations, the loop was repeated two thousand times for each of the three wing sections. Then, a second postprocessor, called POST2.FOR was invoked to change the file of raw elemental stress data into statistical data. Computed for each element was its mean stress, the standard deviation of the stress, and the probability that it would be stressed beyond the maximum allowable stress, given random damage somewhere on the wing. The stress distributions were assumed to be normally distributed random variables thus allowing the probability of failure to be approximated by (1:932)

$$p_f = 1 - .5(1 + c_1 x + c_2 x^2 + c_3 x^3 + c_4 x^4) \quad (2)$$

where: $x = (\nu - \mu)/\sigma$
ν = maximum allowable stress
μ = mean elemental stress
σ = standard deviation of stress

$c_1 = 0.196854$ $c_2 = 0.115194$
$c_3 = 0.000344$ $c_4 = 0.019527$

Once these elemental probabilities were obtained, they were plotted out with SURFER (4), a three-dimensional graphics program. The resulting graphs show the estimated probability of structural failure at an elemental level given that random damage has occurred somewhere in the wing (Figure 5). Figures 6, 7, and 8 show computed probability of structural failure given that damage has occurred somewhere in the root, mid and tip sections respectively. Note that in this model, damage even to the tip section results in increased probability of failure in the root section, Figure 8. This is because damage to any element results in the inability of that element to carry its share of nodal loading. Since that load must be picked up by the surrounding elements, its effect is transmitted through moment resisting joints quite a distance away. Because some of the root elements were highly stressed anyway, the additional stress transmitted to them was sufficient to push them past the failure criterion.

The next step was to investigate the use of SFEA as a tool to increase the reliability, or in this case, the survivability of a structure. Baseline failure probabilities were assembled using 2000 runs per section of the wing, for a total of 6000 runs. Initially, elemental failure probabilities were not known, of course, but were estimated to be in the range of 0.25 to 0.50, given the highly stressed initial conditions in many elements, even without damage.

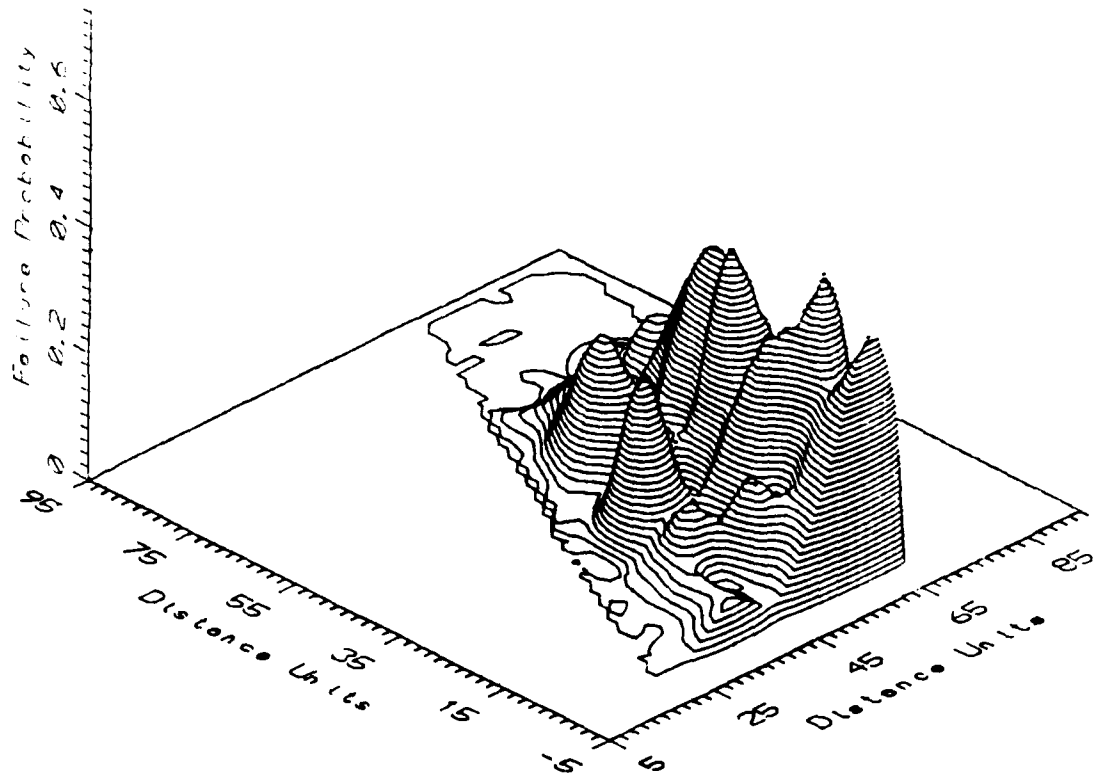


Figure 5. Combined Wing Failure Probabilities; Original Wing

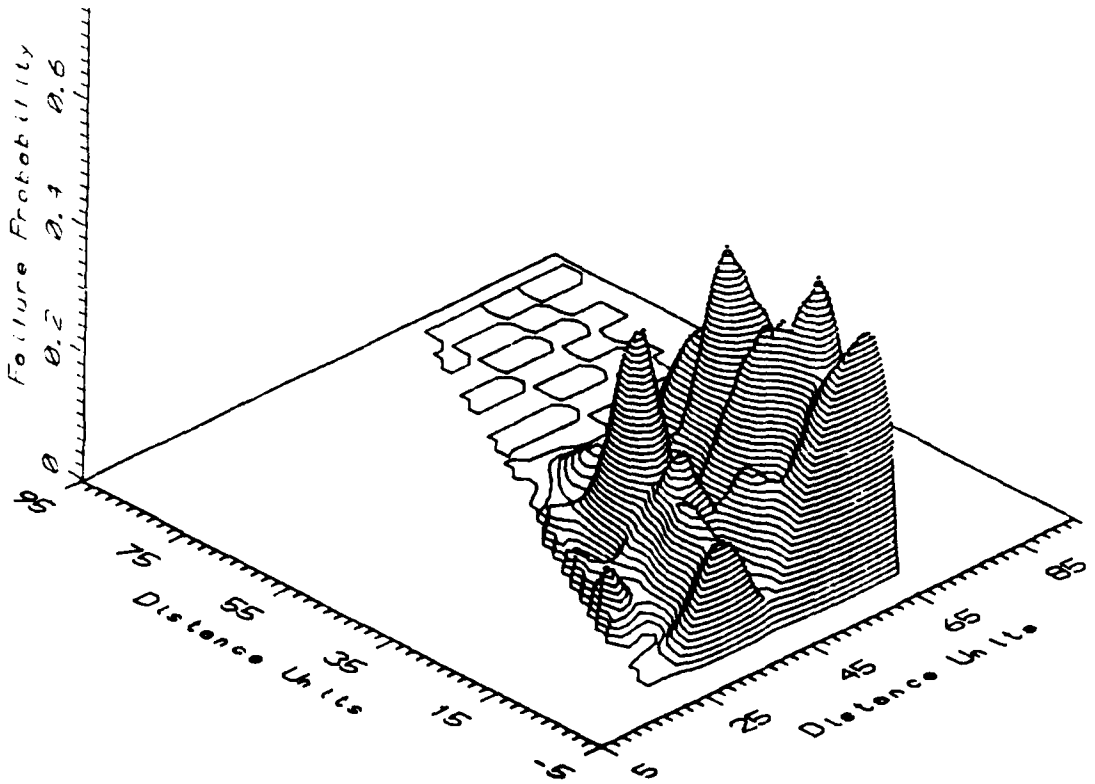


FIGURE 6. Root Damage Failure Probabilities; Original Wing

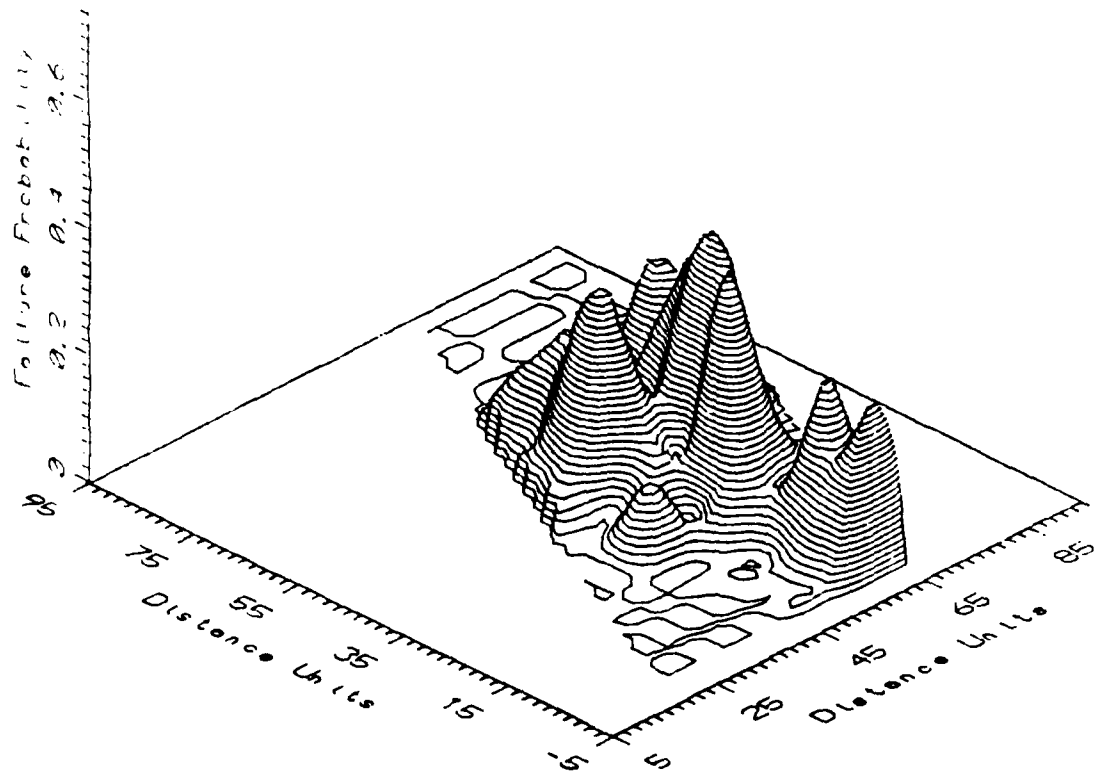


Figure 7. Mid damage Failure Probabilities; Original Wing

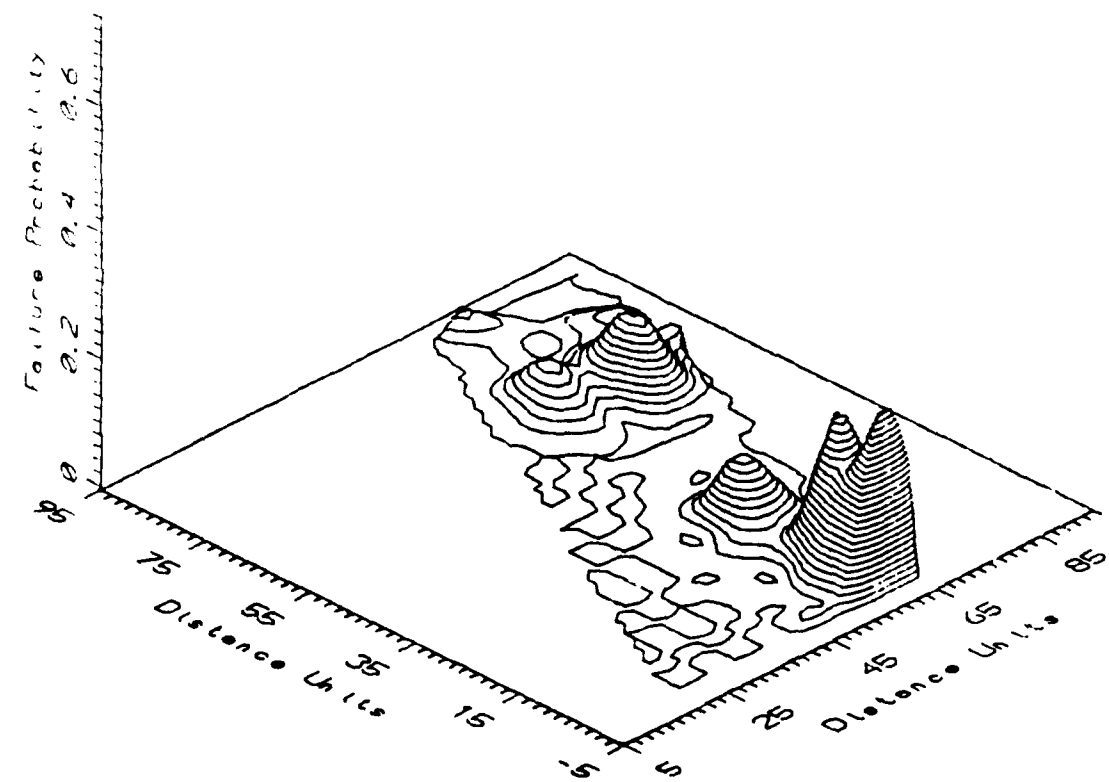


FIGURE 8. Tip Damage Failure Probabilities; Original Wing

The following equation for estimating error in Monte Carlo simulations was used to select the number of runs to establish the baseline (3:291-292)

$$\% \text{ error} = 200 \left[(1 - p_f)/(np_f)\right]^{1/2} \qquad (3)$$

where:
p_f = elemental failure probability
n = number of iterations

The results of this estimator are plotted out in Figure 9 for several selected failure probabilities. Since an error between 5% and 10% was desired, this led to a range of number of runs from 400 to 4800. Two thousand was selected as a good blend of accuracy and economy.

Once baseline statistics were assembled, the two quad elements with the lowest probability of surviving random damage (elements 63 and 64) were strengthened by increasing their thicknesses by 1/64 inch. Then the entire analysis was run again. However, this time through, each section was analyzed only 200 times, instead of 2000. This decreased number of runs meant higher error, up to a maximum of 100% for a failure probability of 0.02, but that increase would be offset by a corresponding decrease in the failure probabilities through the design modifications. Thus even a 100% error with a failure probability of 0.02 still yields a small failure probability range (0-0.04).

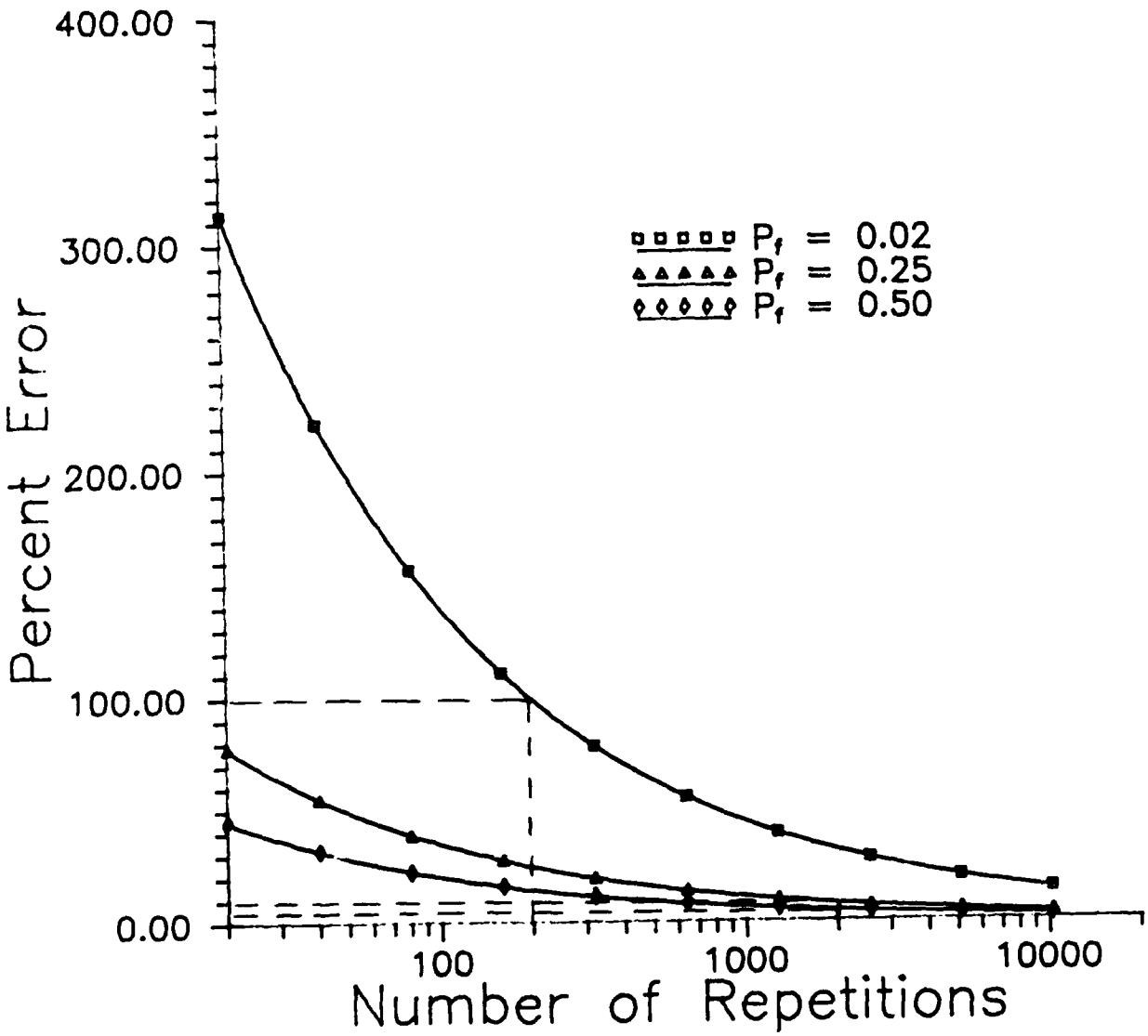


FIGURE 9. Percent Error For Monte Carlo Simulations

The results of this series of runs were analyzed. The unexpected result was that while the additional thickness in the two strengthened elements raised the likelihood they would survive a hit, the additional stiffness imparted by the extra thickness caused new failures in near by elements. Figure 10 graphs the failure probabilities of the combined wing (all three sections susceptible to damage) for the unimproved wing and the wing with elements 63 and 64 strengthened. Note the very low failure probability in elements 63 and 64 (right corner of graph) and the slight increase in failure probabilities in elements 45, 46, 53, 54, 55, and 56 (the peak next to 63 & 64). This increase in failure probabilities is more evident when the wing is sectioned off. Figure 11 shows a comparison of the original vs. the strengthened wing for damage confined to the root area. Not much change is evident, but in Figures 12 and 13, showing comparisons of the mid and tip sections, the increase in failure probabilities is obvious. Effects of stress telegraphing are plain, especially with tip damage. The newly strengthened elements are failure-free while the adjacent element has been pushed into failure. This result means that strengthening parts of a structure can actually lower its resistance to failure under certain conditions, such when damage is confined to a certain area. The next iteration led to strengthening these neighboring elements, another round of 600 total runs, followed by the probability calculations. This time, about twenty neighboring elements were strengthened. A trend was emerging. While adding strength to selected elements still resulted in

neighboring elements experiencing more failures, the trend was flattening out, resulting in more elements being more survivable. Figure 14 compares the initial improvement (round one) combined wing with the secondary improvement (round two) combined wing. Overall, the failure probabilities have decreased. Figure 15 compares the root section of the round one and round two wing. Again there is a general decrease in failure probabilities. Figure 16 compares the mid section of the two wings. While failure probabilities are still relatively high in the mid section of the round two wing, wing-breaking stress is no longer being shifted into other sections. Figure 17 compares the tip sections of the two wings. Note that even though failure probabilities increased for one element in the mid section of the round two wing, there were no longer any root section failures.

By the end of the third round of strengthening, most elements had reliabilities of over ninety percent, and those that didn't were only down to about 85%. Note, in Figure 18 comparing the combined wing of round two with that of round three, the general improvement in failure probabilities. Improvement is evident when comparing each of the three wing subsections with the round two version, in Figures 19, 20, and 21.

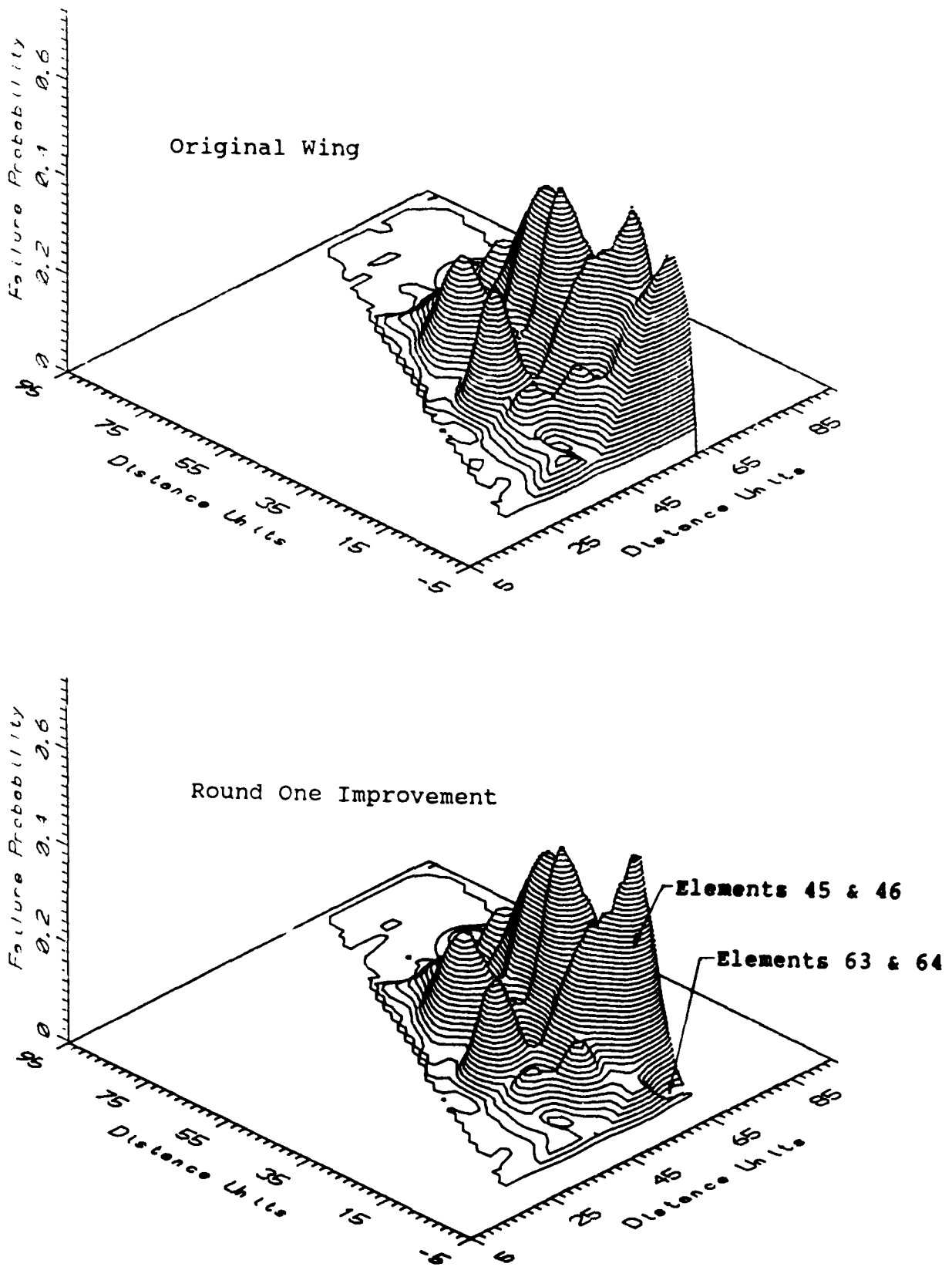


FIGURE 10. Combined Wing Failure Probabilities; Original vs. Round One Improvements

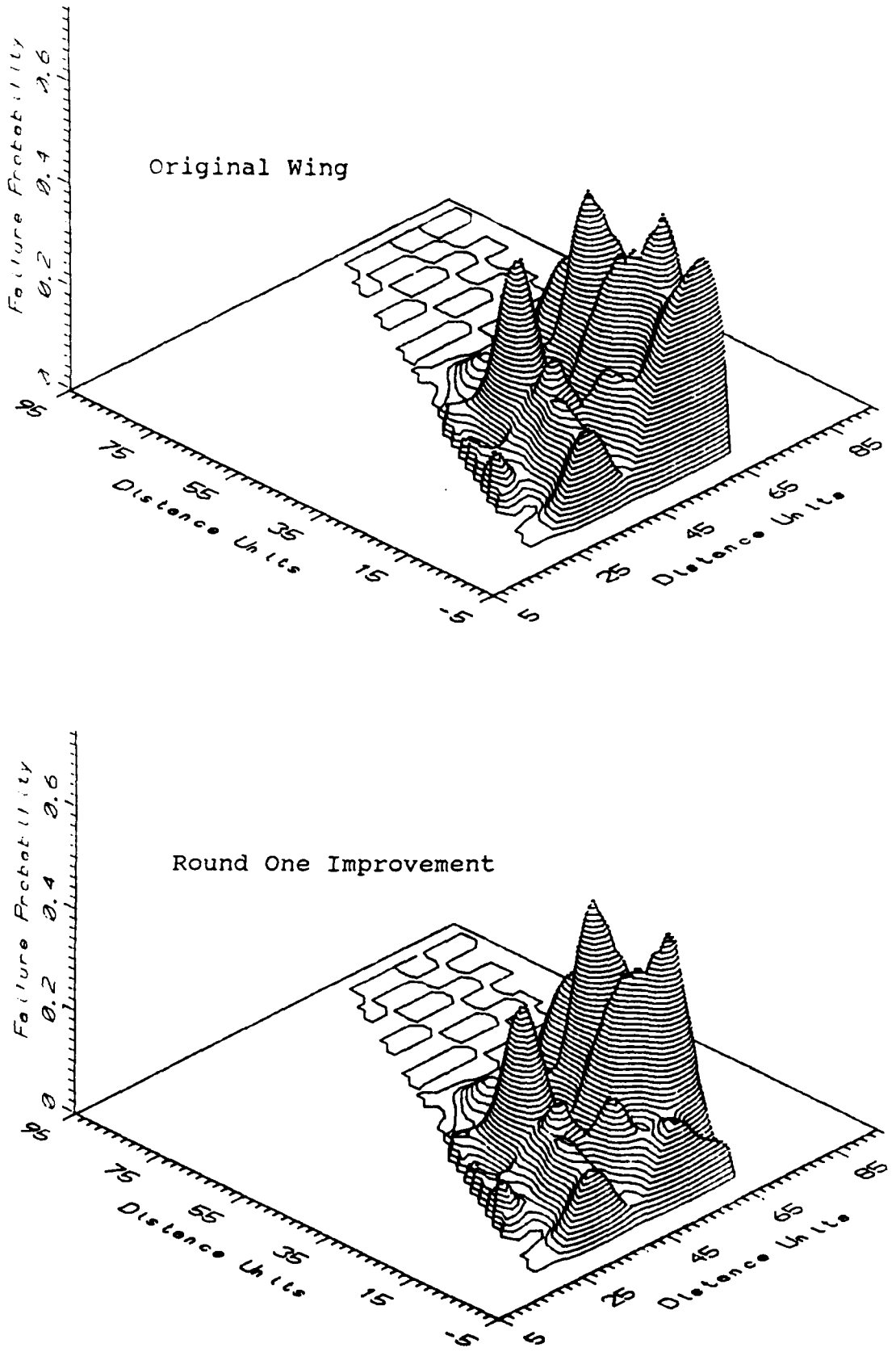


FIGURE 11. Root Damage; Original vs. Round One Improvements

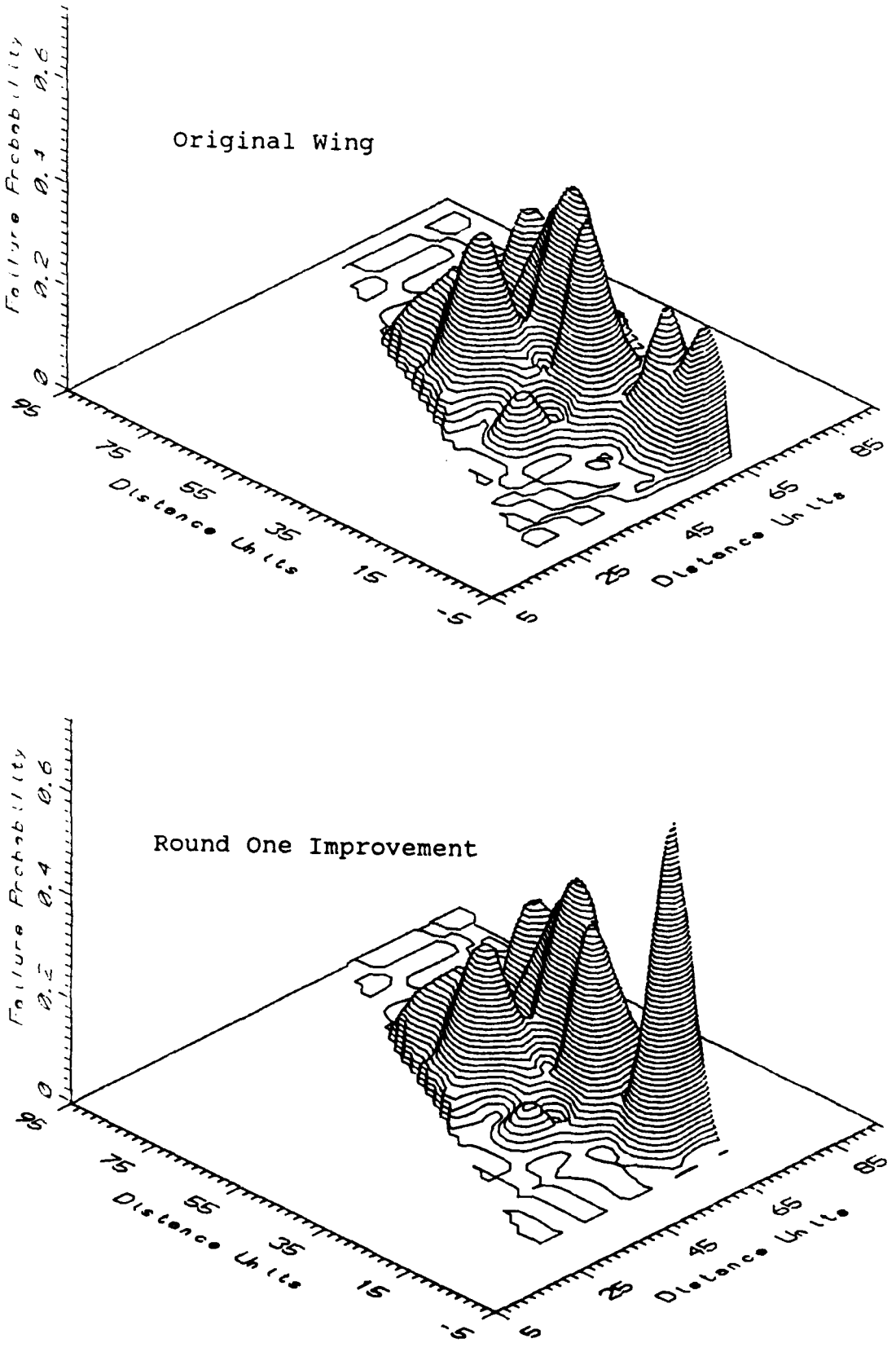


FIGURE 12. Mid Damage; Original vs. Round One Improvements

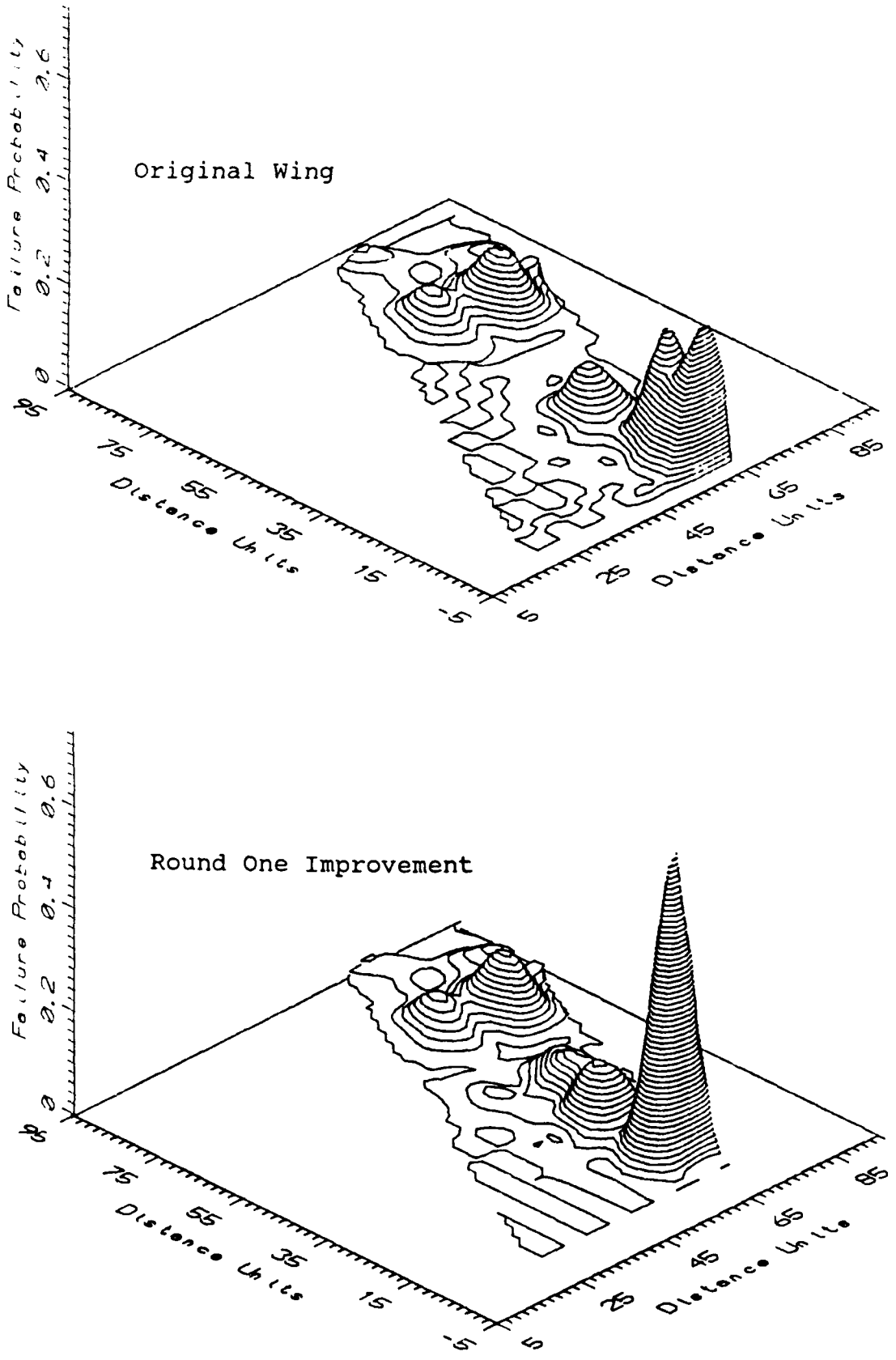
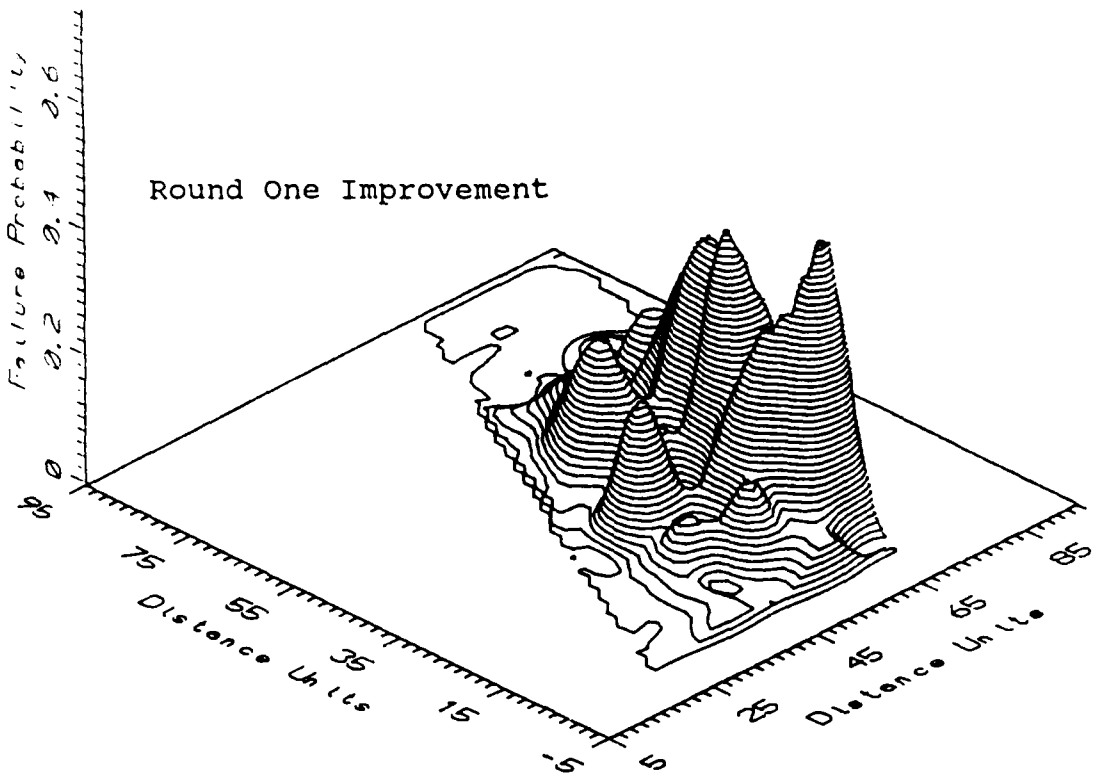


FIGURE 13. Tip Damage; Original vs. Round One Improvements



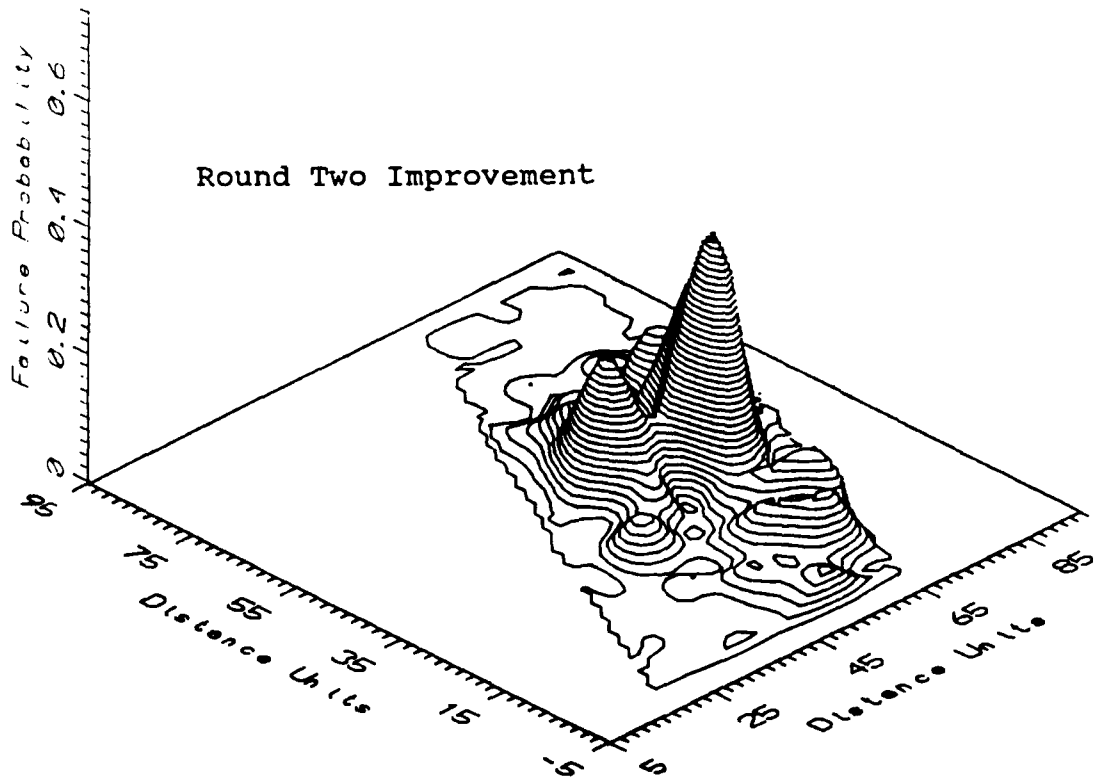


FIGURE 14. Combined Wing; Round One vs. Round Two

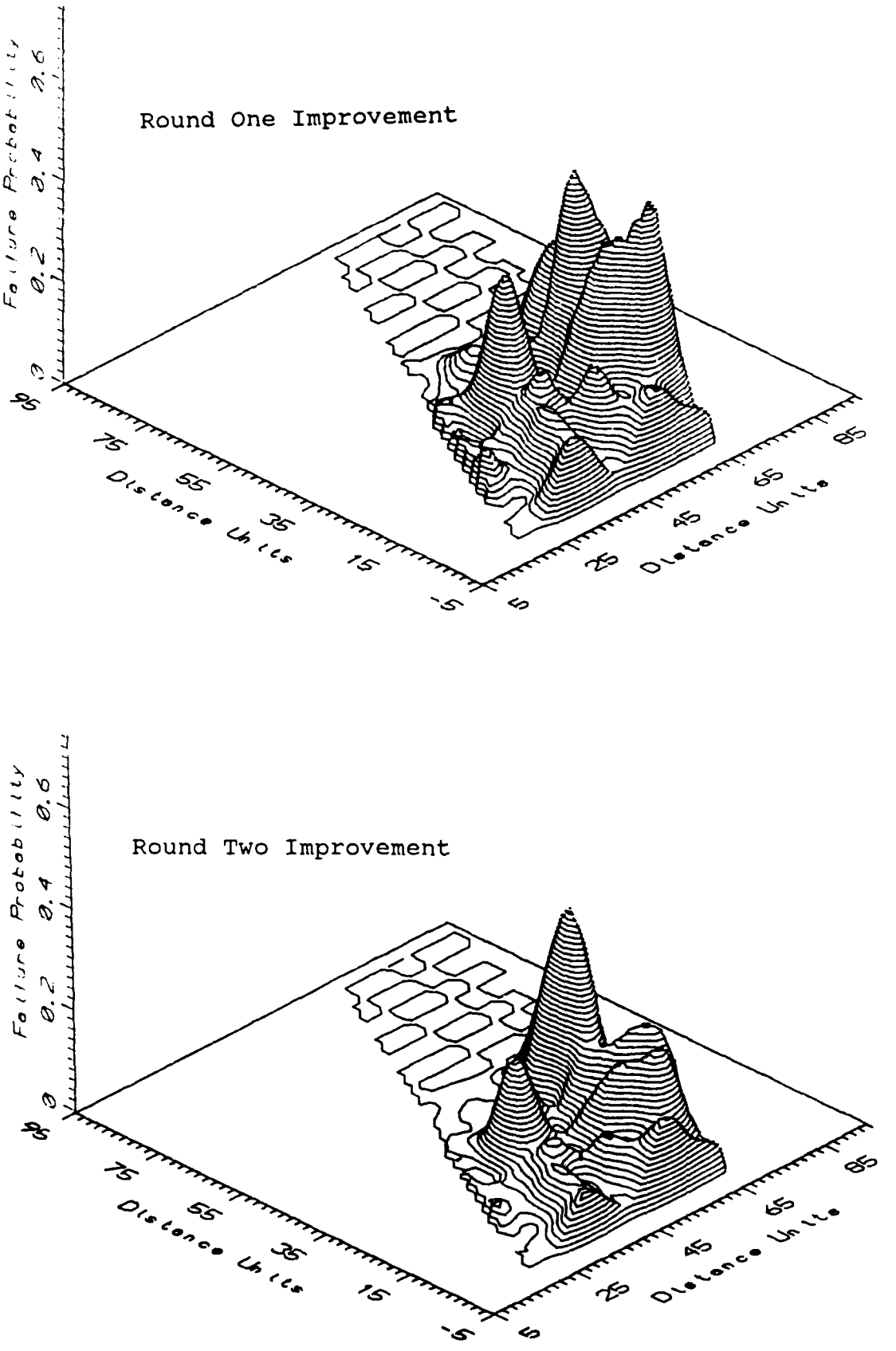
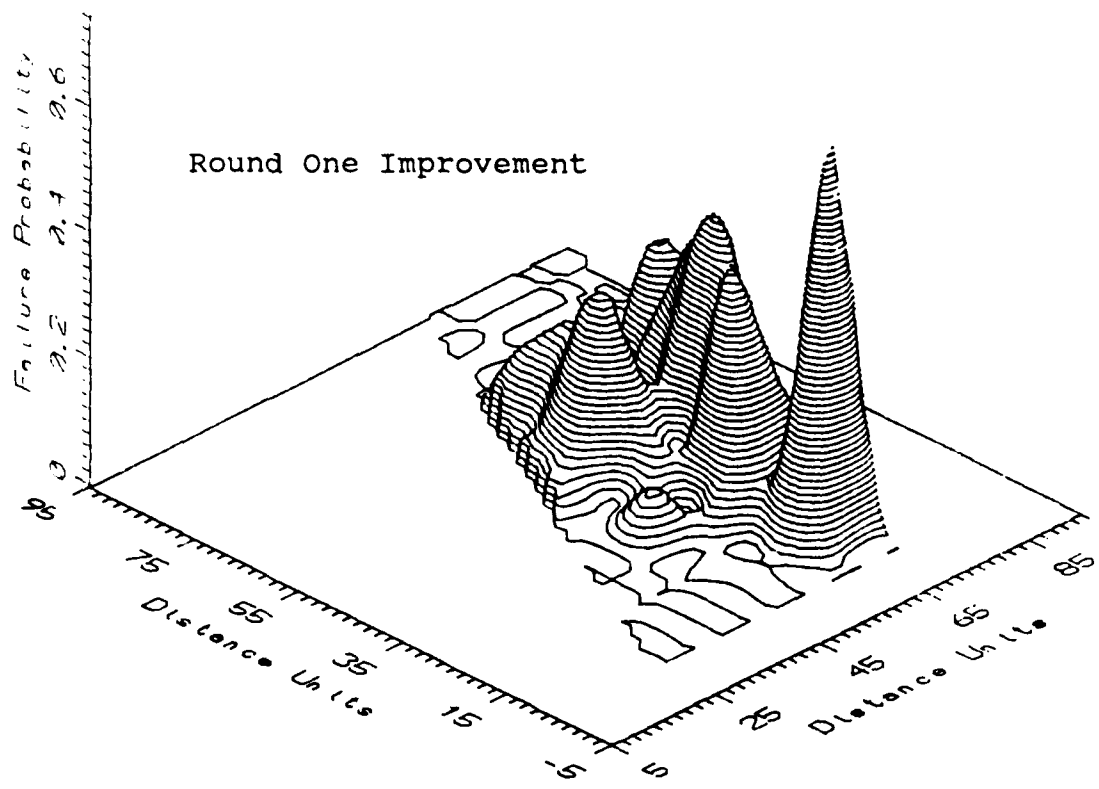


FIGURE 15. Root Damage; Round One vs. Round Two



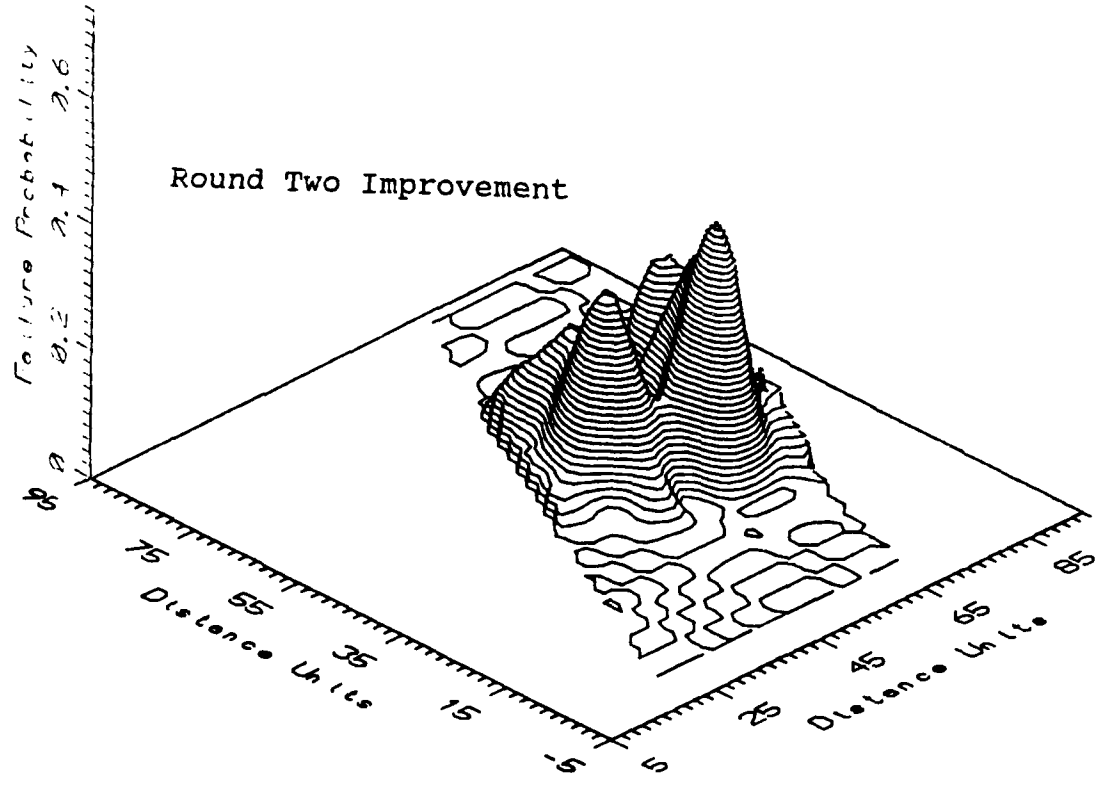
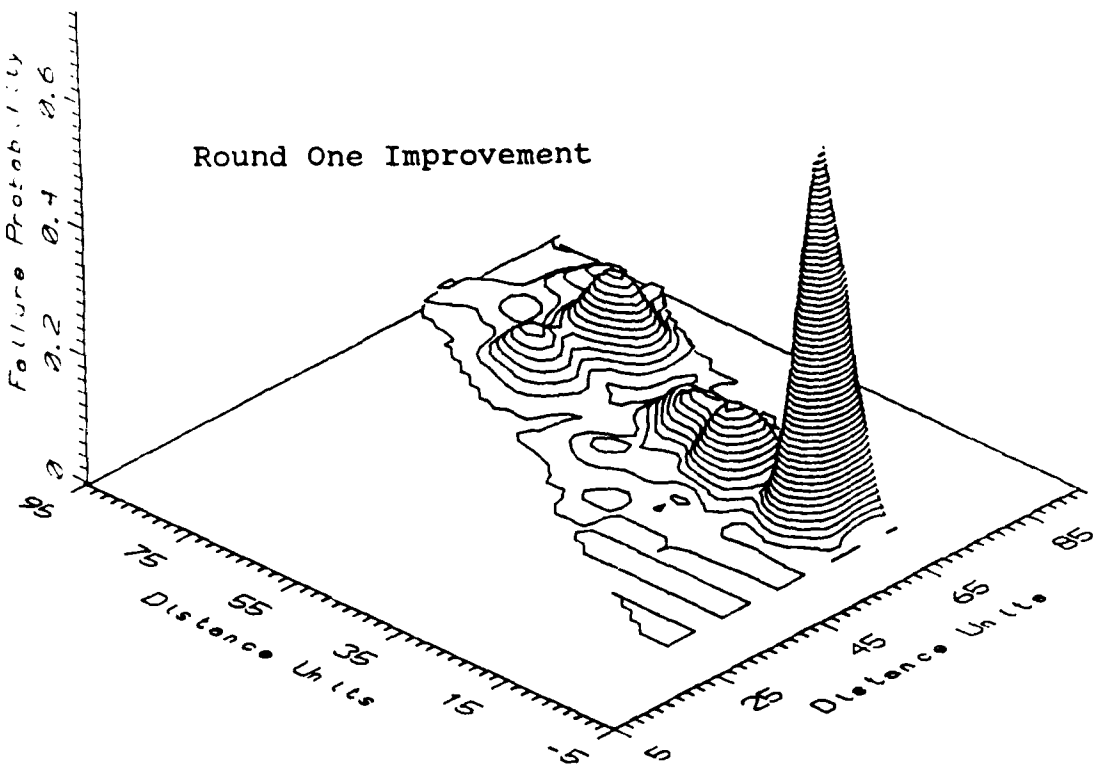


FIGURE 16. Mid Damage; Round One vs. Round Two



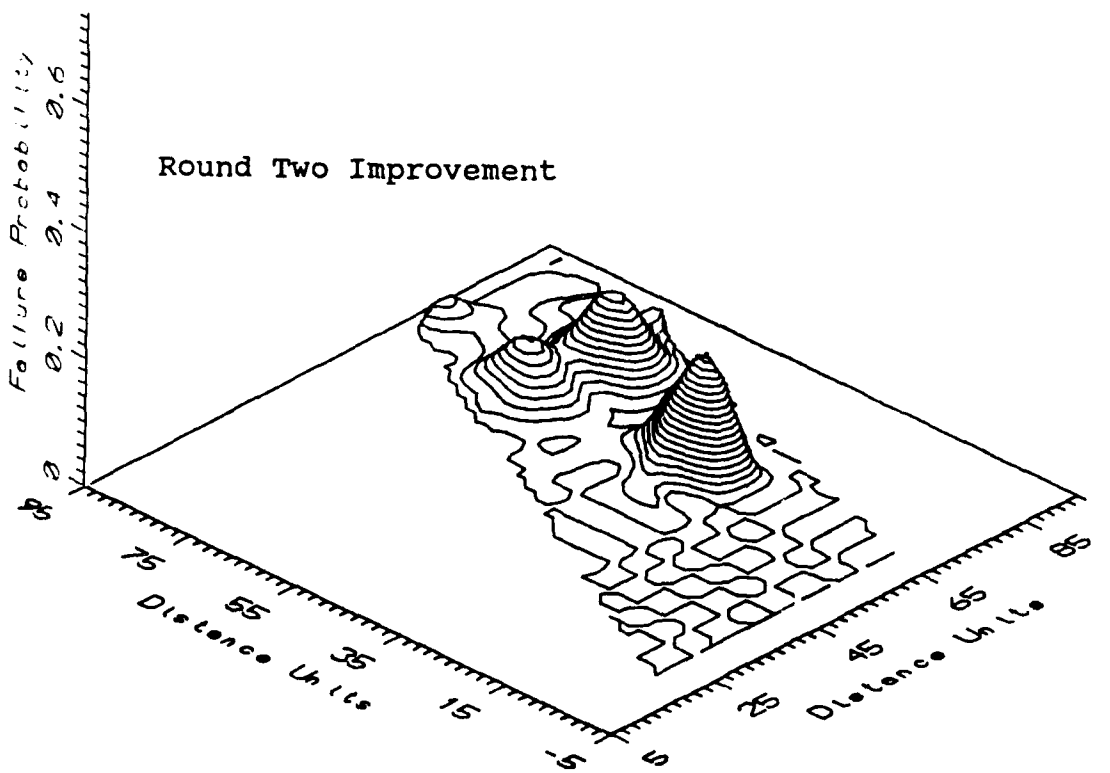
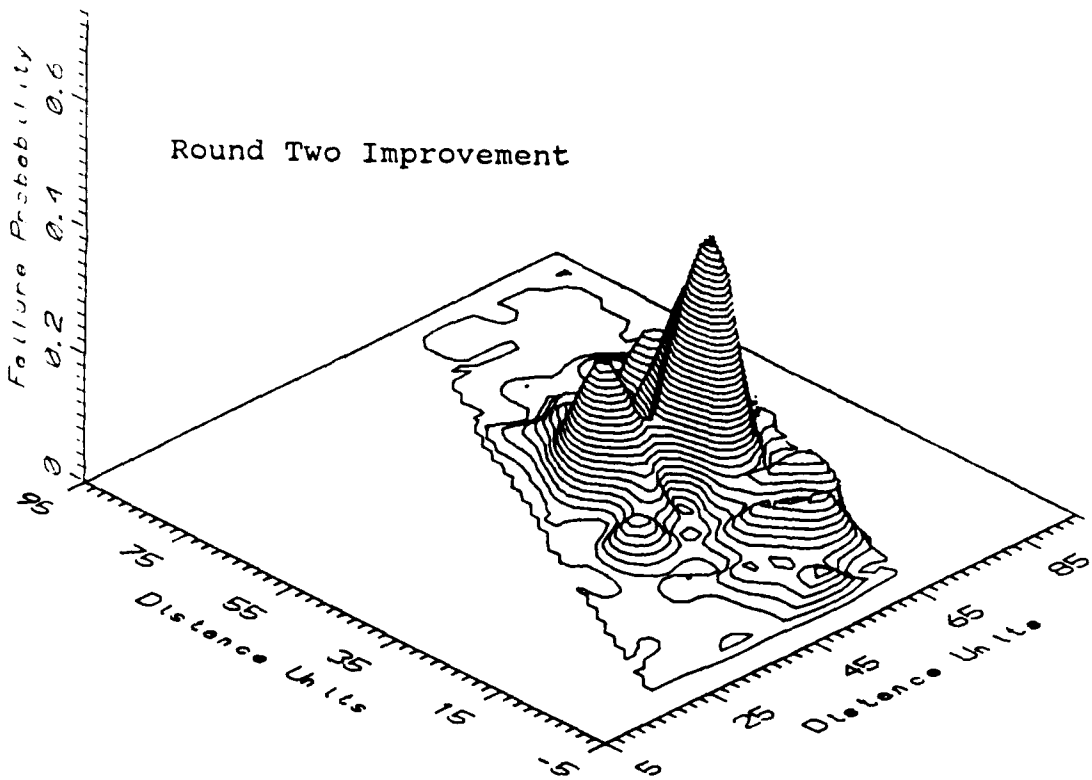


FIGURE 17. Tip Damage; Round One vs. Round Two



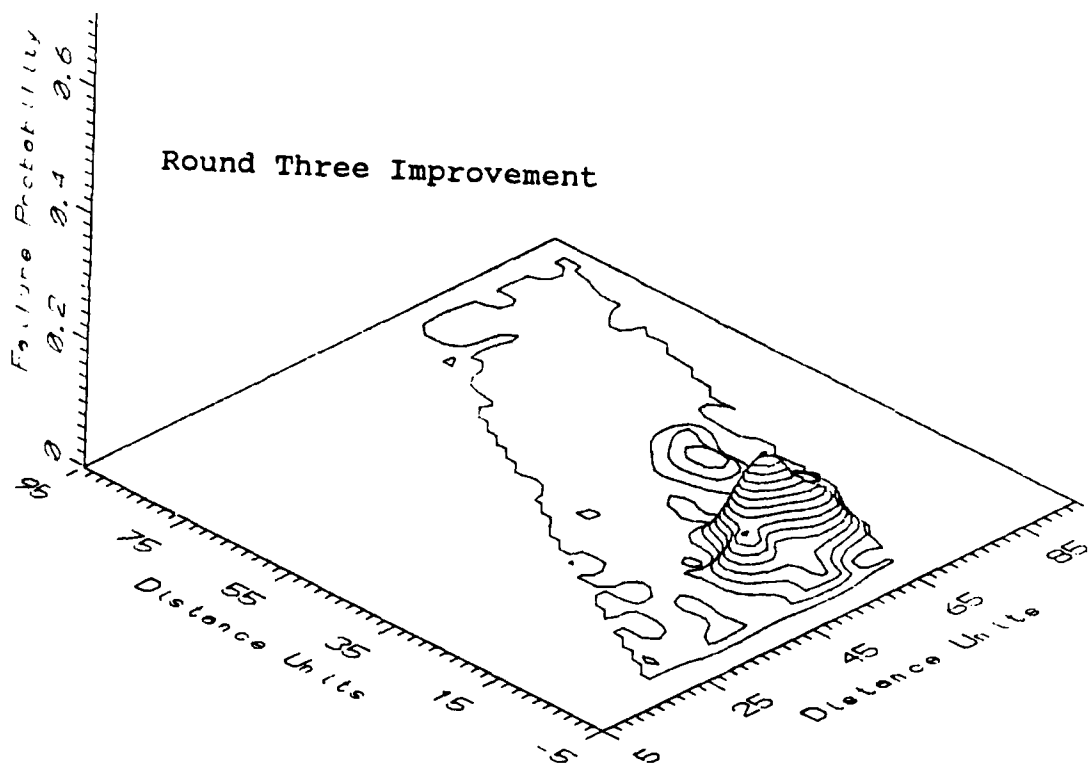
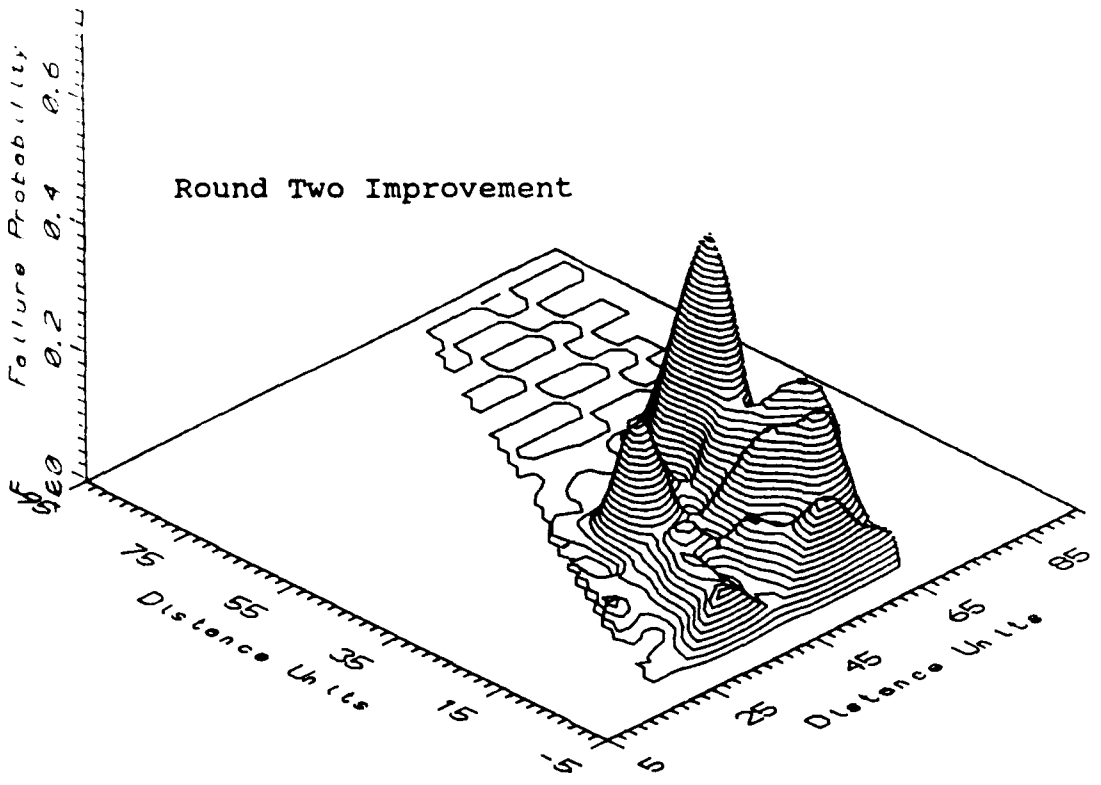


FIGURE 18. Combined Wing; Round Two vs. Round Three



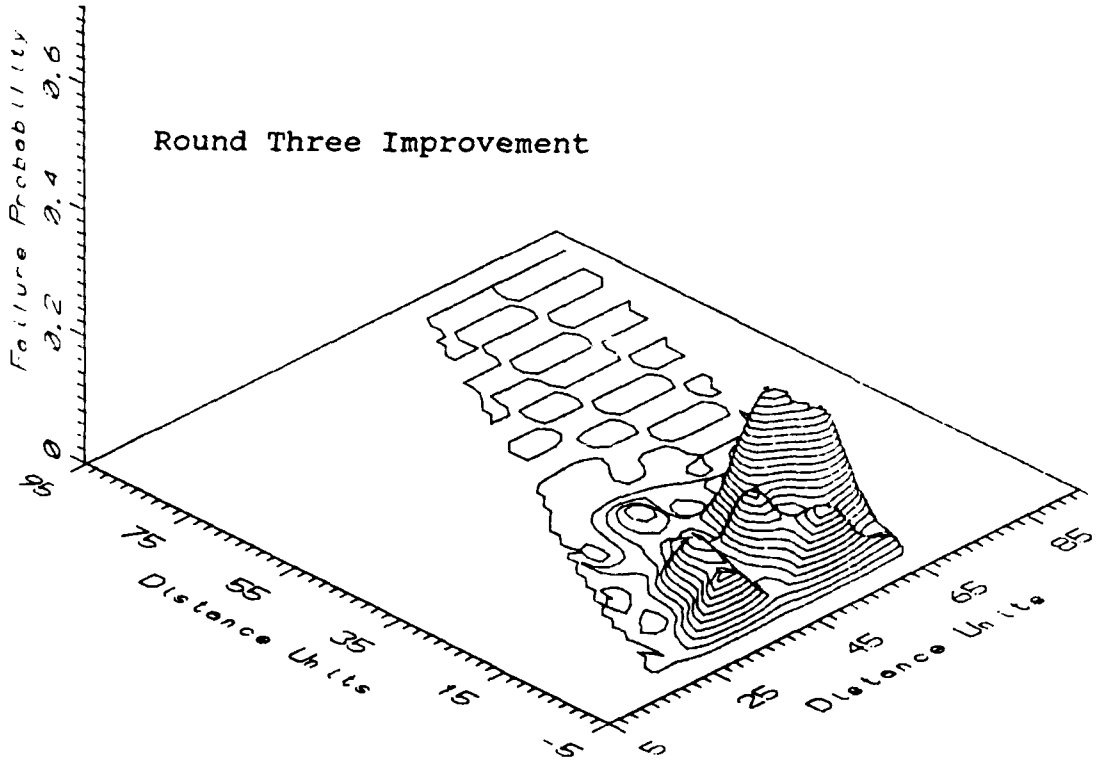
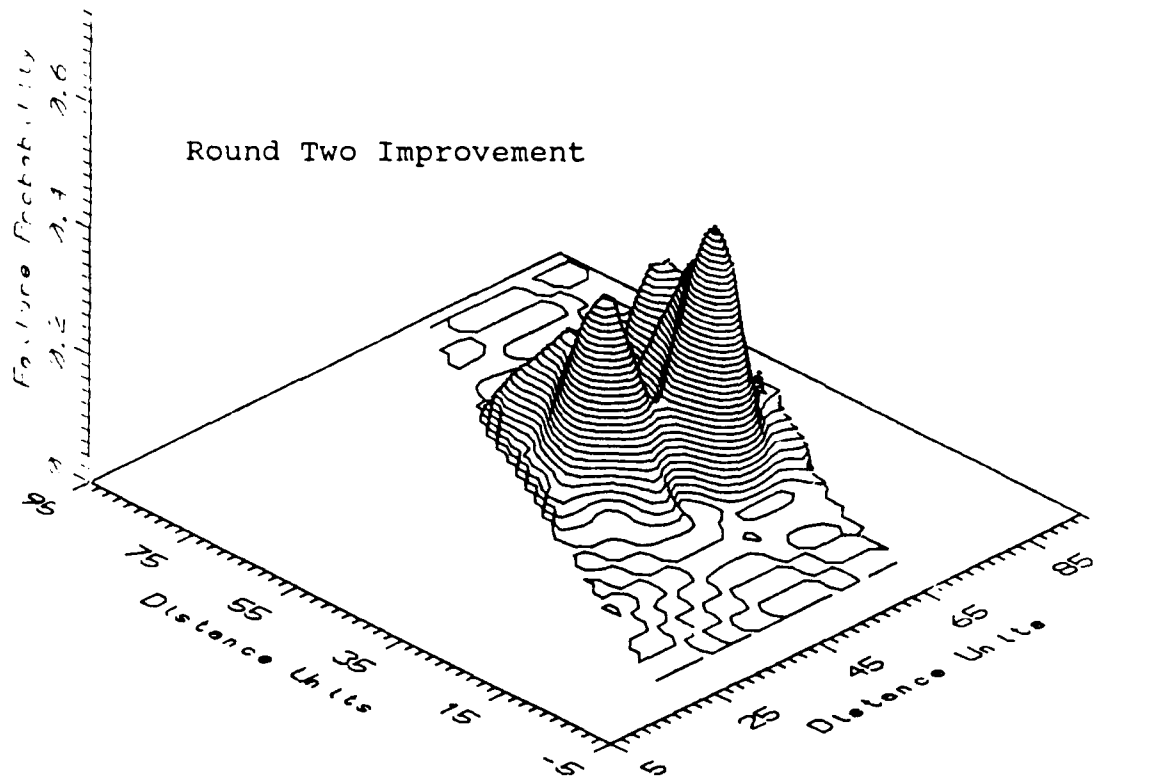


FIGURE 19. Root Damage; Round Two vs. Round Three



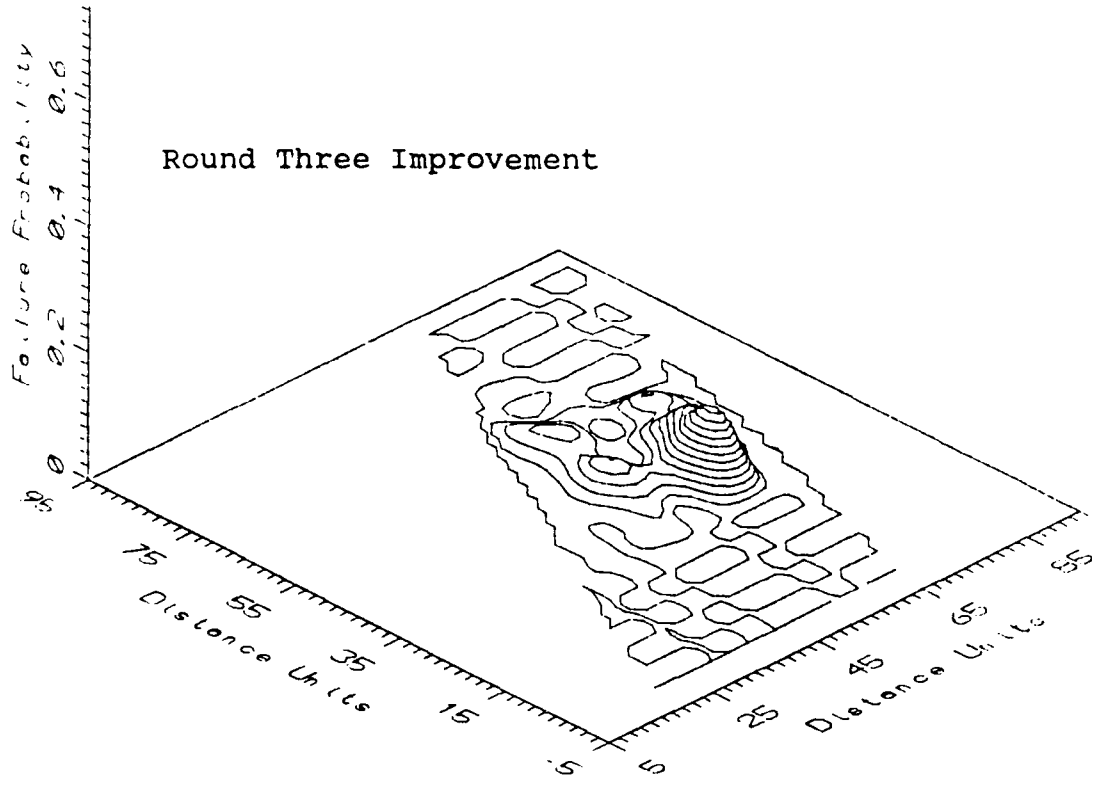


FIGURE 20. Mid Damage; Round Two vs. Round Three

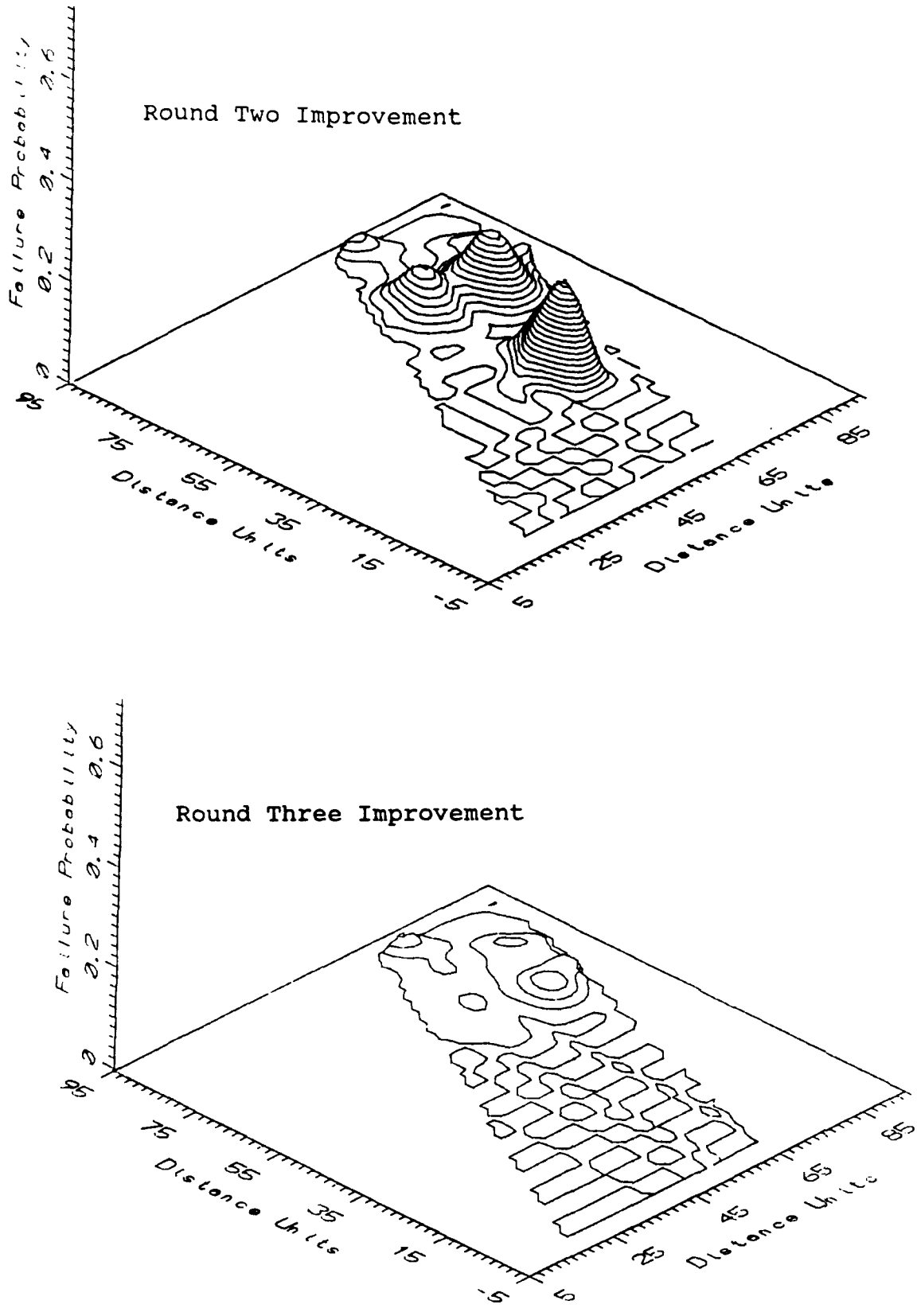


FIGURE 21. Tip Damage; Round Two vs. Round Three

The final step was to trim some weight from those elements whose survivability was never in question. Obviously, these elements were in a lightly stressed state as their reliabilities were always 1.0. With this final adjustment, the survivability of almost every element was greater than 90%, as opposed to the original design, wherein many elements had reliabilities in the 60-70% range. Interestingly, trimming the weight from these lightly stressed elements helped improve the reliability of more heavily stresses neighboring elements, in much the same way that adding weight to heavily stressed elements in round one decreased the reliability of adjacent elements. On the downside, however, this weight savings did result in some of the tip elements' failure probabilities increasing. For purposes of this demonstration, this final step results in a wing referred to as "best". This is not meant to imply that this is as reliable as the wing can be, but only that it was as good as this demonstration made it.

Figure 22 shows the element numbers of those elements whose strength was affected in each round of design modification. All incremental changes were 1/64 inch in thickness. Numbers in paraentheses represent decreases in thickness to save weight. The cost of this increase in survivability was estimated at a mere 4% increase in weight. (See Appendix E for more details of this estimation.) This was estimated by computing the solid volume of the structural material of the original wing model and the new structural solid volume imparted by the design modifications.

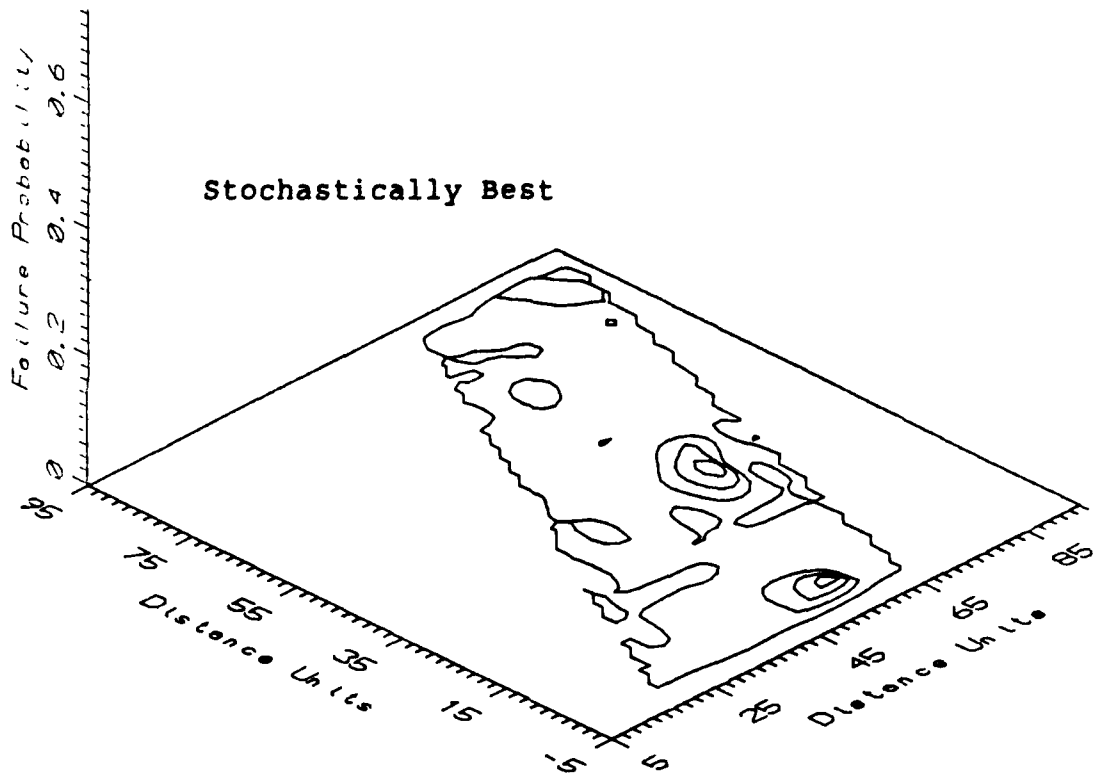
LIST OF STRENGTHENED ELEMENTS

Round One	Round Two	Round Three	Best
63, 64	45, 46	19, 20	(5, 6)
	53, 54	21, 22	(7, 8)
	55, 56	27, 28	(9, 10)
		29, 30	(11, 12)
		31, 32	(13, 14)
		33, 34	(15, 16)
		35, 36	(17, 18)
		37, 38	(19, 20)
		39, 40	(25, 26)
		43, 44	(41, 42)
		45, 46	
		51, 52	
		53, 54	
		57, 58	

Parentheses indicate elements lightened

FIGURE 22. Elements Improved, Round by Round

In contrast, to get about the same increase in survivability using the factor of safety approach, a safety factor of 1.25 had to be applied to get reliabilities in approximately the same range as the best wing obtained by the stochastic method. It may be interesting to note the usual range of safety factor used for structural aluminum is 1.65-2.34, with an average about 2.0 (7:510). While the safety factor for a highly weight-controlled structure, such as an aircraft, could be expected to have a fairly low safety factor, it is not normally below 1.5; unmanned structures, such as missiles can go as low as 1.25 (6). Figure 23 shows the combined wing comparison between the best wing obtained stochastically and the wing improved by the 1.25 factor of safety. Overall, the stochastic wing has better reliabilities. Figure 24 and 25 show the root and mid section comparisons. The stochastically best wing has low failure probabilities in the damaged root and mid sections along with a few very low failure probabilities in the tip area, as shown by the wavy lines at the tip. The factor of safety wing had higher failure probabilities in the damaged sections, but essentially zero failure probabilities outside the damaged sections. Only when damage was confined to the tip section, Figure 26, do we see the reverse of this. The factor of safety wing had very low failure probabilities all across the wing, whereas the stochastically best wing had tip failures rates as high as 0.1. Note, however, a safety factor of 1.25 means that weight was increased by 25%! The stochastic method of analysis resulted in a more reliable structure, with a weight savings of more than 20%!



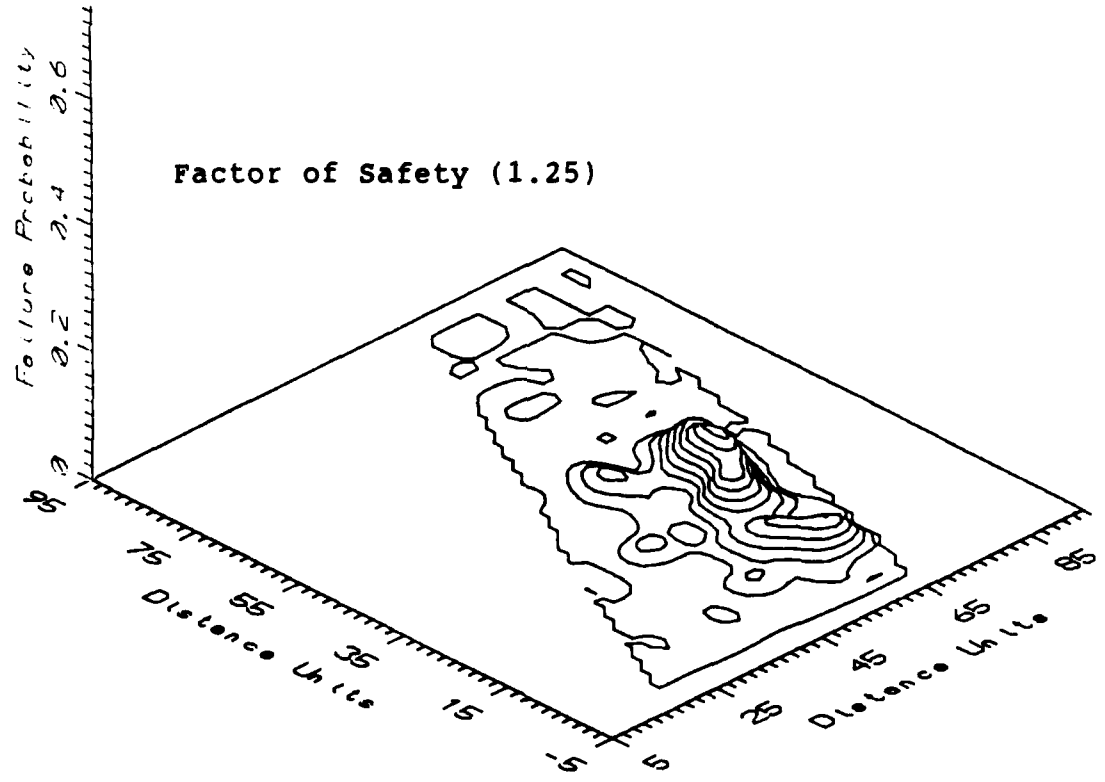
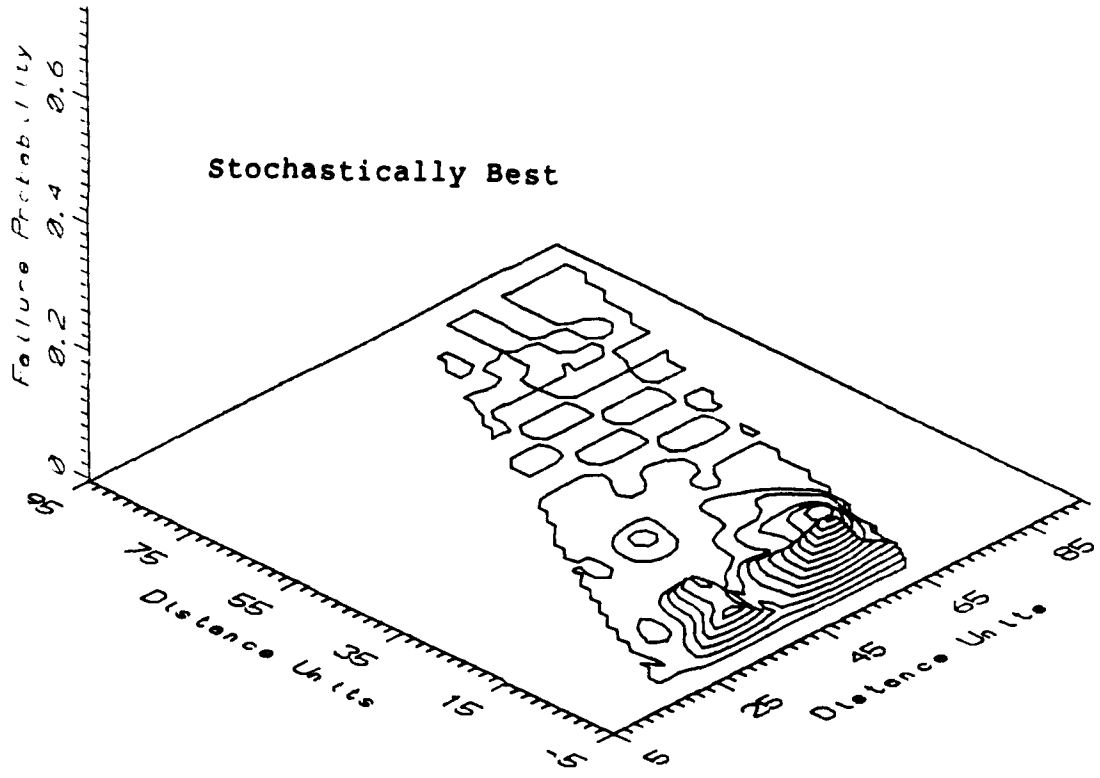


FIGURE 23. Combined Wing; **Stochastically Best vs. Factor of Safety**



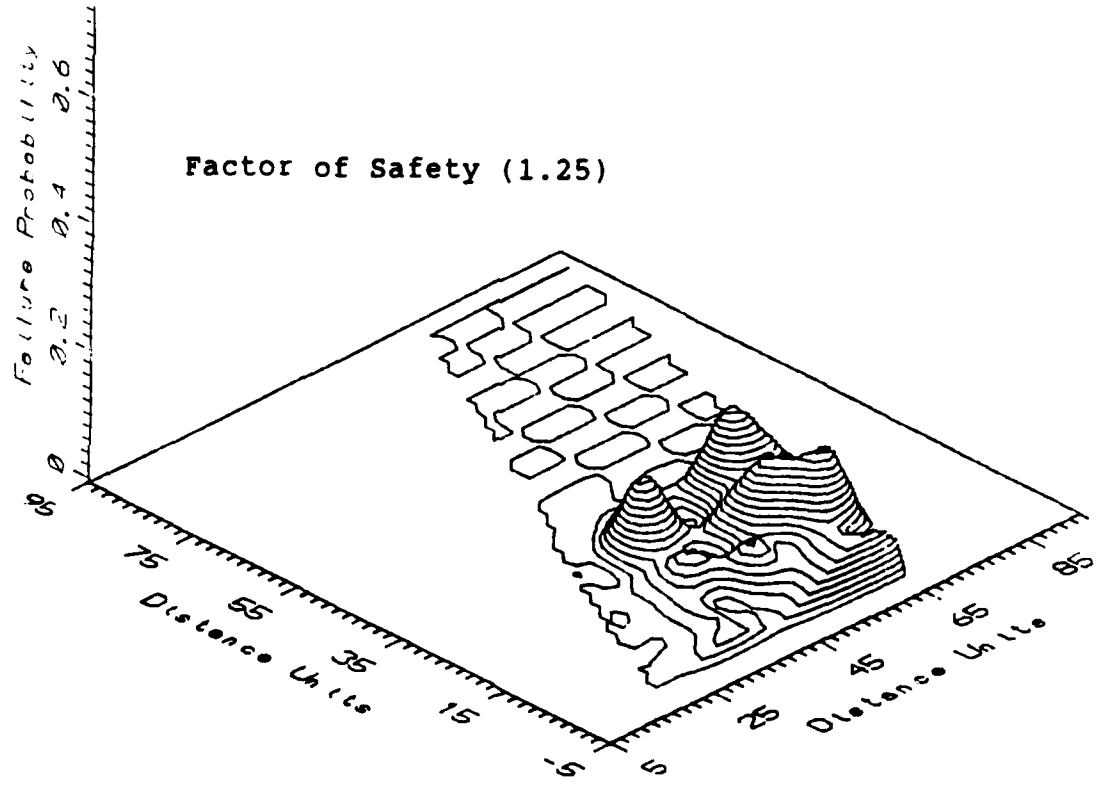
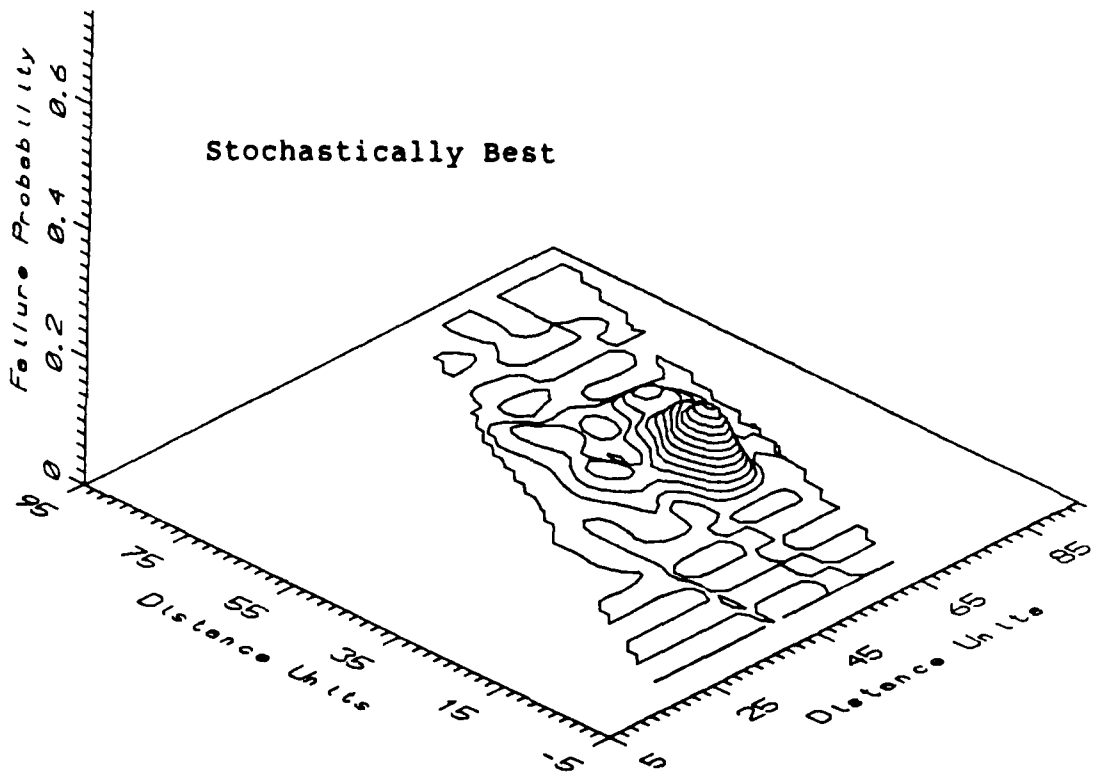


FIGURE 24. Root Damage; Stochastically Best vs. Factor of Safety



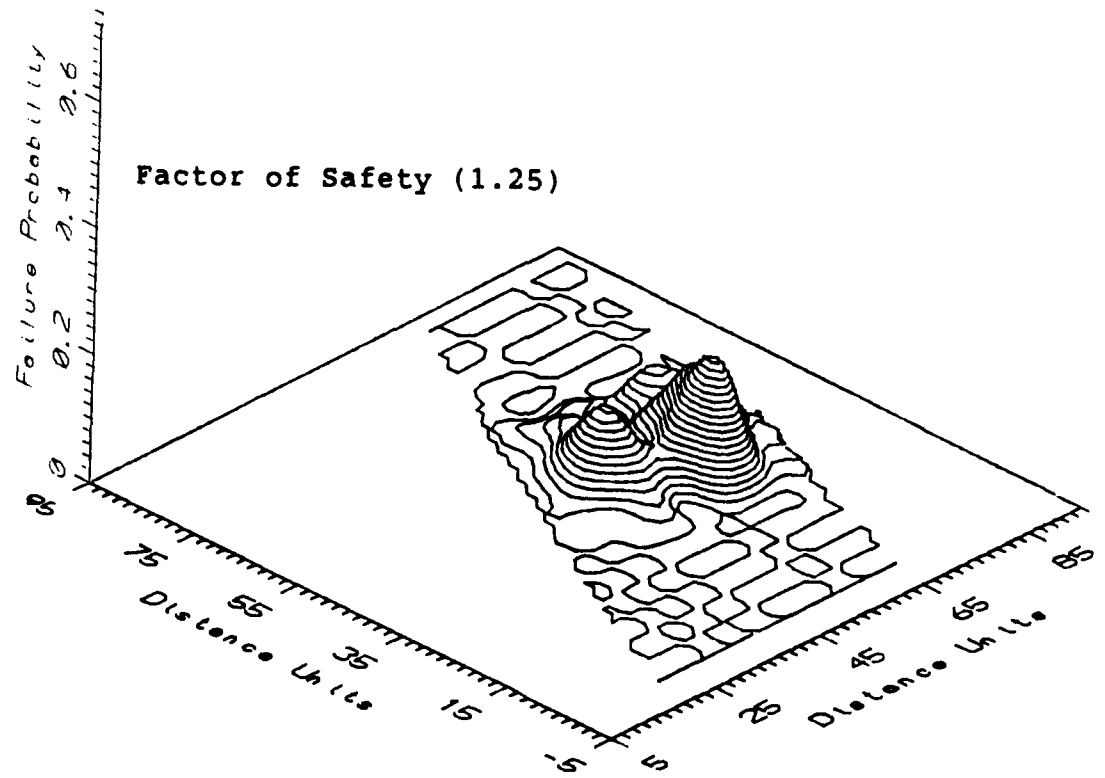
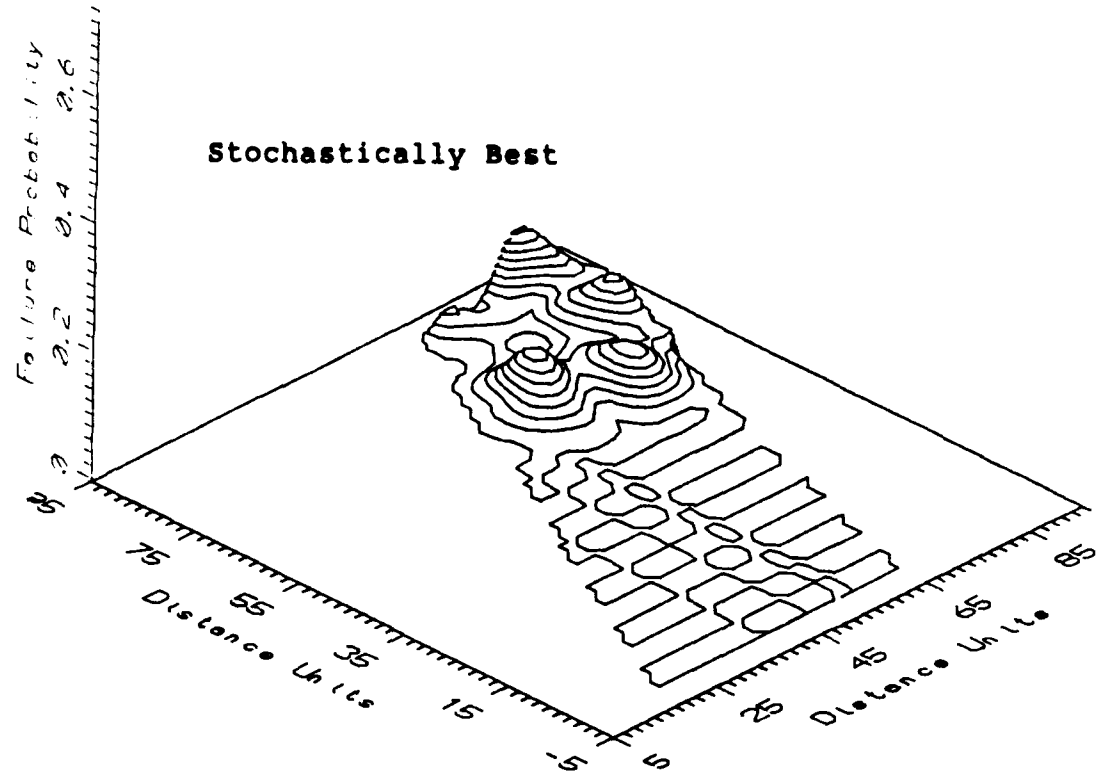


FIGURE 25. Mid Damage; Stochastically Best vs. Safety Factor



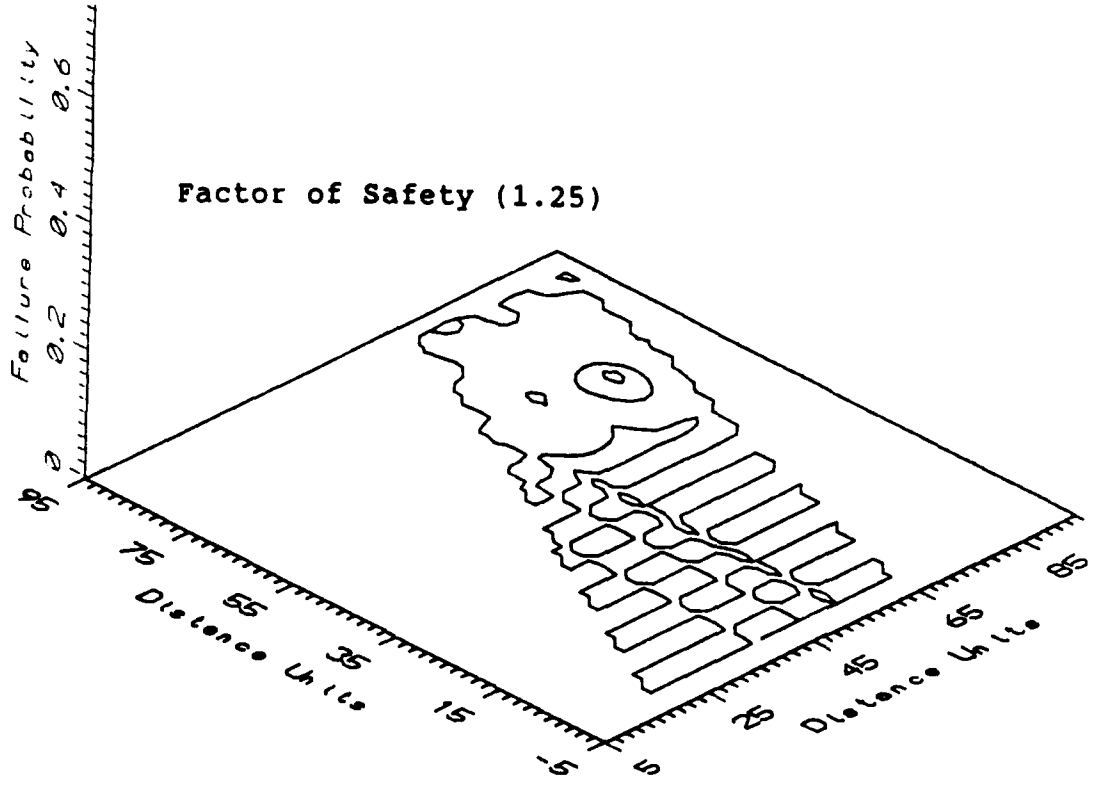


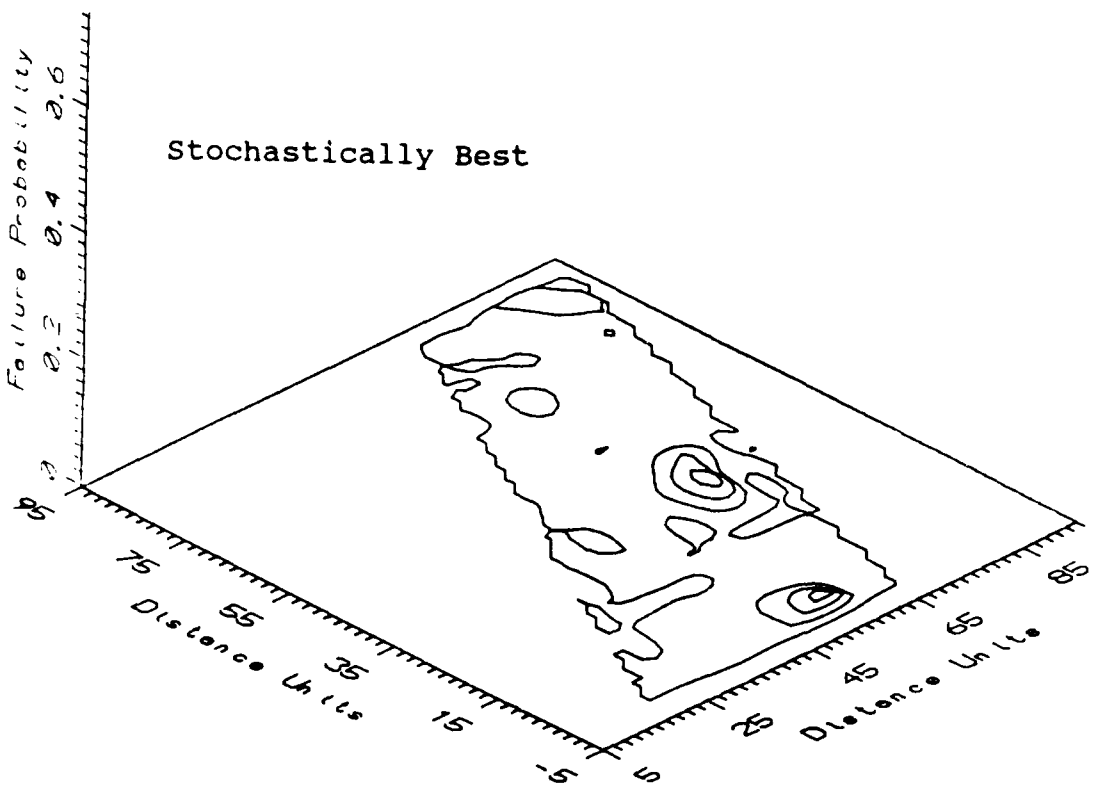
FIGURE 26. Tip Damage; Stochastically Best vs. Safety Factor

The final step in this demonstration involved comparing two different optimization schemes. One scheme used the optimizing capability of ASTROS. In an iterative procedure, ASTROS adjusts the thickness of the model's structural members to exactly resist the applied loads (which are adjusted to the desired factor of safety.) ASTROS is one of the very few finite element programs available with this built-in function; virtually all other programs rely on the designers ability to interpret the output data and update the model accordingly. Thus, ASTROS represents the state-of-the-art in optimization codes.

The other method was to continue the stochastic process already started, using a a short program, POST3.FOR, to update the elements' thicknesses based on failure probabilities.

First, ASTROS was instructed to optimize the model to a safety factor of 1.25. The estimated weight gain over the original model was about 22%, while the failure probabilities compared to the stochastically best wing (at 4% weight gain) are shown in Figure 27. Next, ASTROS optimized to a safety factor of 1.5, with a resultant weight gain estimated at about 30% over the original wing. The failure probabilities of this wing were essentially zero out to the fourth and fifth decimal place. Then the previous stochastic programs were run under a new batch file along with the POST3.FOR program acting as a crude optimizer. (The term crude is applied as POST3.FOR did not allow for any lightening of the elements--it would only

strengthen them.) Essentially, the stochastic optimizer worked like this. At the end of the standard 600 runs, POST2.FOR was called to generate elemental failure probabilities. Then, POST3.FOR would scan that probability list looking for "large" failure rates. (What constituted large changed as the failure probabilities converged to zero. For the first couple of runs, large was 0.1; later, it was 0.001. The number could be changed within POST3.FOR while the optimizer was running.) It would then increment the thickness of the failure prone elements, and the whole procedure would start again, for ten iterations. At this point, the stochastically optimized wing's failure probabilities very closely matched the ASTROS optimized wing with the safety factor of 1.5. The weight gain for this stochastically optimized wing was about 25%. Thus, for essentially the same reliability, the stochastic method of design utilizing a crude optimizing scheme gave a design 10% lighter than that produced by one of the best optimization programs available.



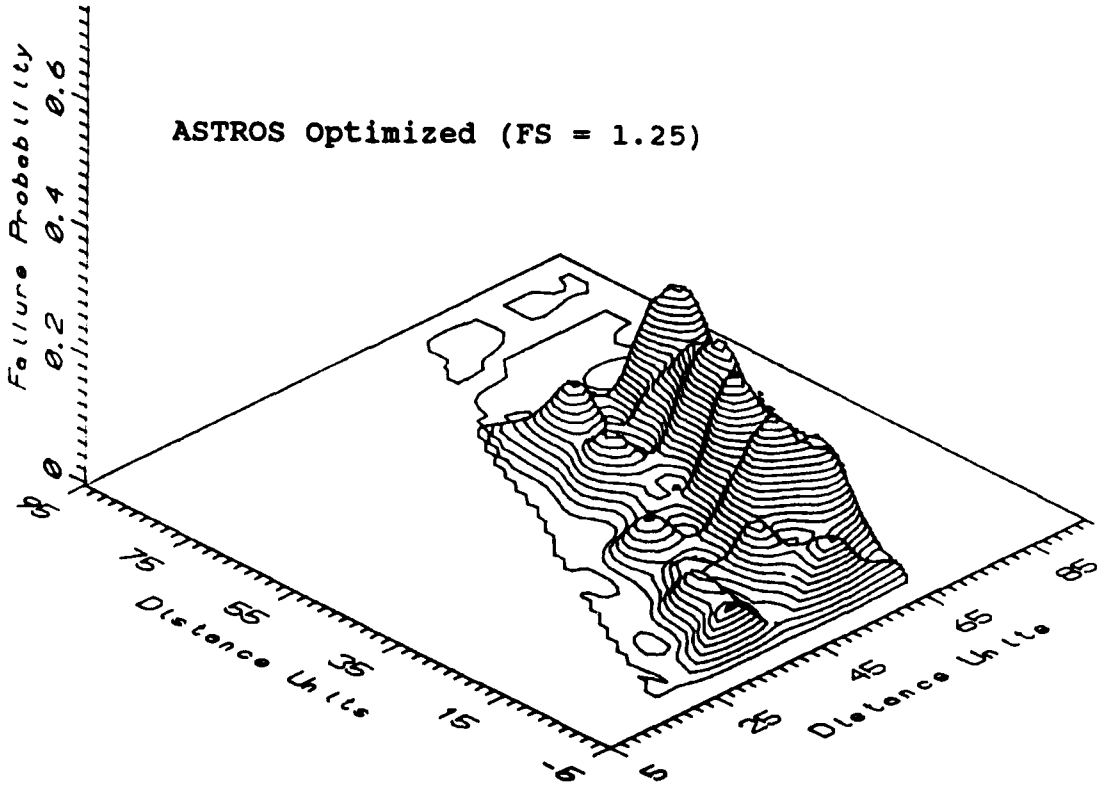


Figure 27. Combined Wing. Stochastically Best vs. ASTROS Optimized (FS = 1.25)

Conclusions

Stochastic Finite Element Analysis is a tool which may be used by designers to attempt to improve the reliability or the weight of a structure. Its use involves some limitations and drawbacks in comparison to traditional methods of design using finite element analysis and factors of safety. Principally, the limitations are in the designer's ability to adequately capture probability distribution functions of potential failure modes and then to encode them into a computer digestable algorithm. The principal drawback is the potential for using very large amounts of computer time.

In this demonstration, the limitations were overcome by using simple probability distribution functions to model the hazards. Random variables were restricted to just two: damage location and nodal loading. The random variable of damage was restricted to only one or two structural elements that were easy to select. More complex modes of damage would involve commensurately more complex programming, especially if multiple damage sites and cascading damage effects are considered. The random variable associated with nodal loading was confined to an easily modeled normal distribution. Only one loading condition was considered here, that of supersonic flight. Many other high stress loading conditions exist, such as high-g maneuvers, heavily loaded munitions pylons, even shock wave loading from nearby explosions. Conceivably, a wing optimized for one loading condition could be a terrible

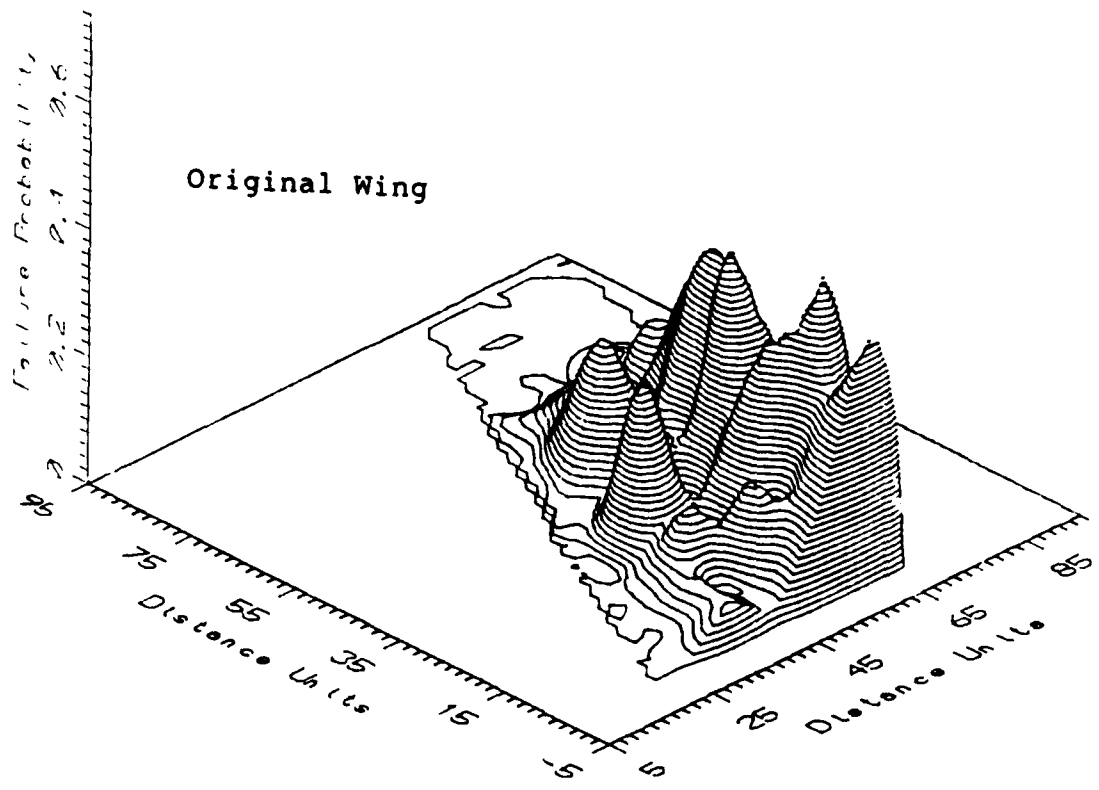
performer for another. However, since structural design is a study in compromise, a solution, representing improvement over the traditional approach, should be almost always achievable.

The drawback of using vast amounts of computer time could be insurmountable. Baseline analysis of the Intermediate Complexity Wing took a week of CPU time on a three mip minicomputer, almost 16 days real time, and several weeks human time. Manipulation of the resulting enormous data files was very slow. Subsequent 600 iteration runs took eighteen hours, real time.This simple demonstration used at least fifteen of them, not including false starts, mistakes, and debugging runs. A very large, complex structure, such as a bomber wing, utilizing thousands of nodes, could take a week for just one run. Access, therefore, to extremely powerful, inexpensive computers is essential to successfully and economically apply the method of stochastic finite element analysis to any large, complex structure.

Nevertheless, given reasonably accurate hazard probabilities and access to sufficient computer resources, this method can be a valuable to in providing a lighter, more reliable structure. Figures 27 and 28 trace the reliability growth of a wing originally designed using traditional finite element analysis methods. As elements of the wing were selectively strengthened, the wing's failure probabilities were systematically reduced to a fairly uniform low rate. Figure 29 compares the stochastically strengthened wing to the wing

strengthened through the factor of safety. While both are highly reliable, the factor of safety wing is more than 20% heavier then the stochastic wing.

When ASTROS, a state-of-the-art finite element program, was called upon to produce an optimized wing based on a safety factor of 1.5, it yielded a wing of extremely high reliability. The stochastic algorithms developed for this demonstration also yielded a wing of similar reliability, but with a weight savings of 10% over the ASTROS optimized wing.



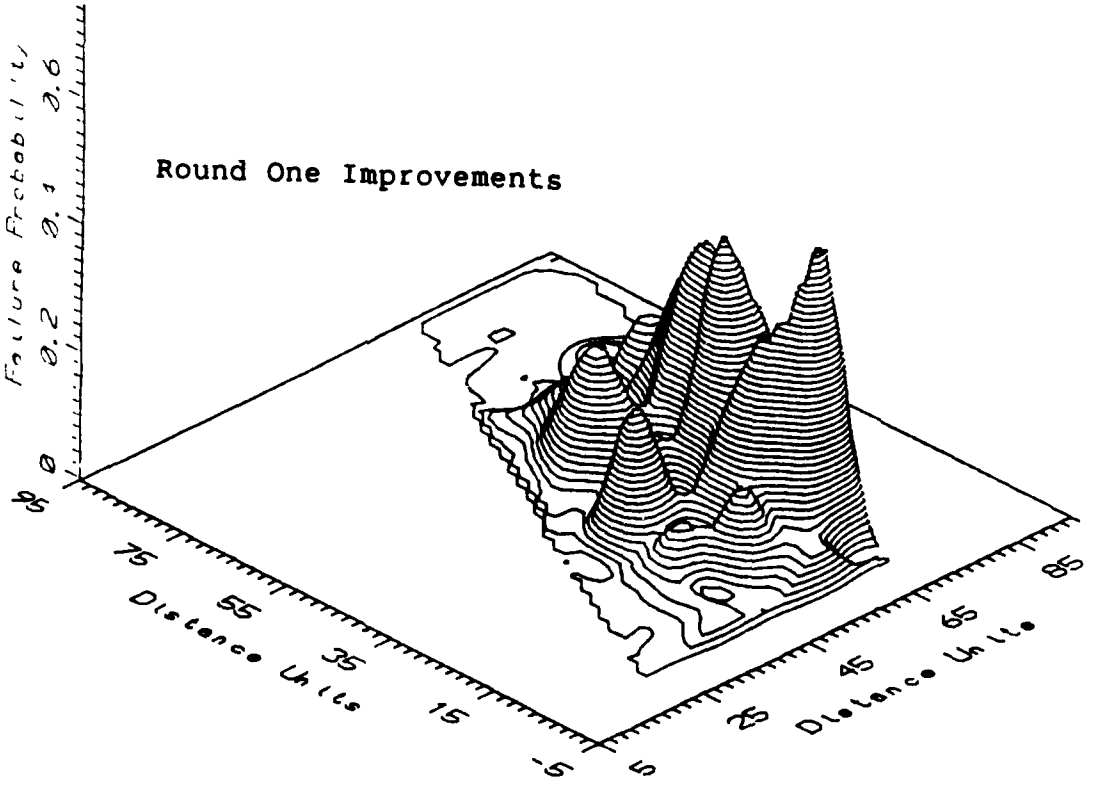


FIGURE 28. Combined Wing. Original Wing vs. Round One Improvements

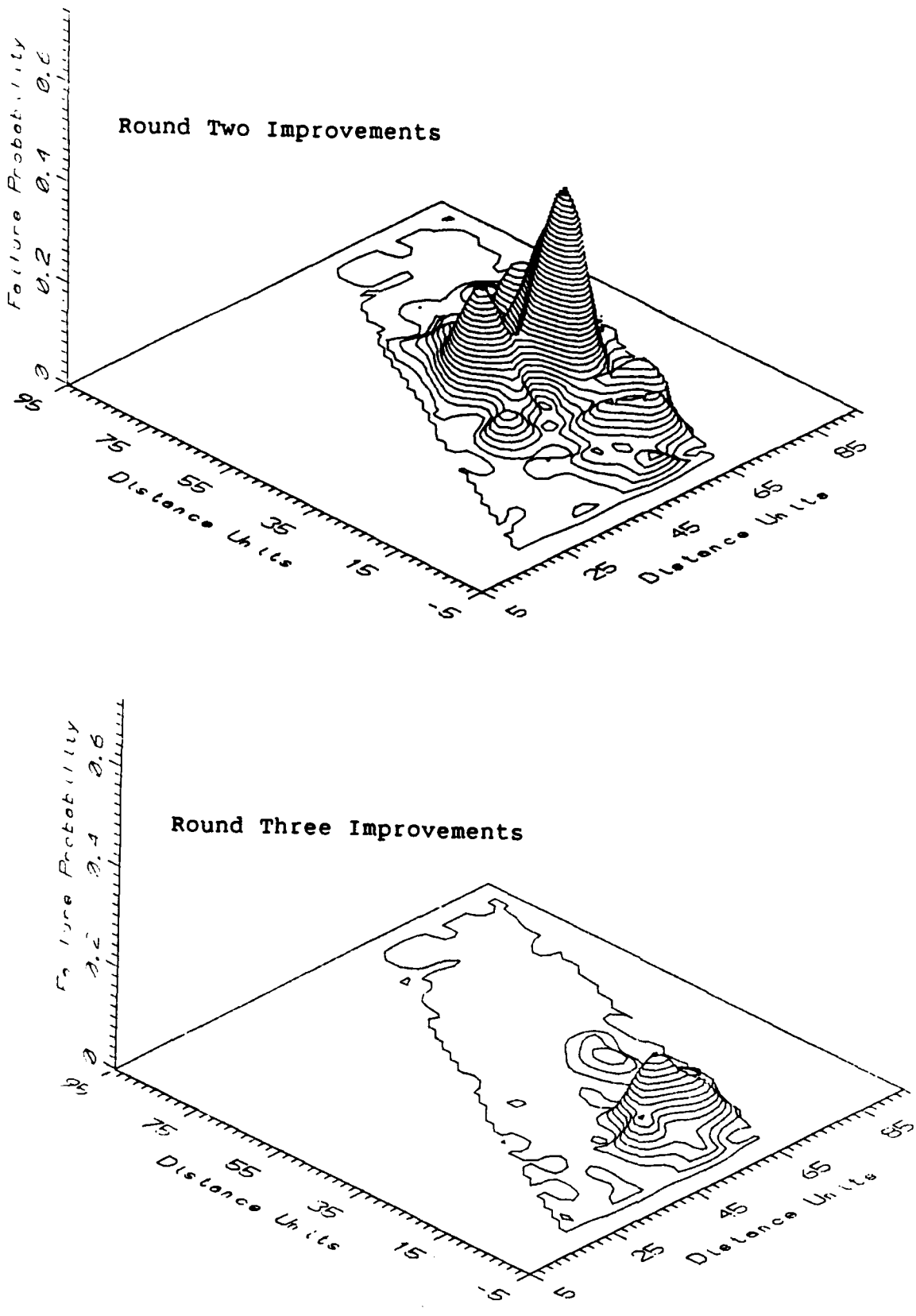
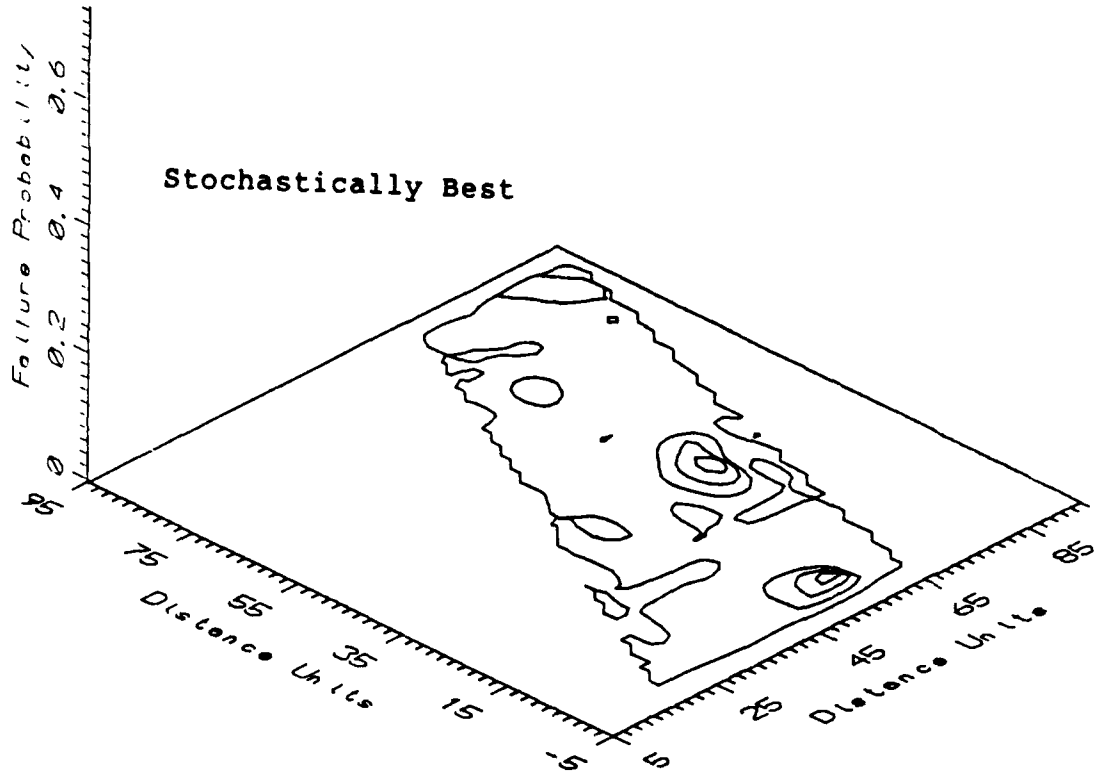


FIGURE 29. Combined Wing. Round Two Improvements vs. Round Three Improvements



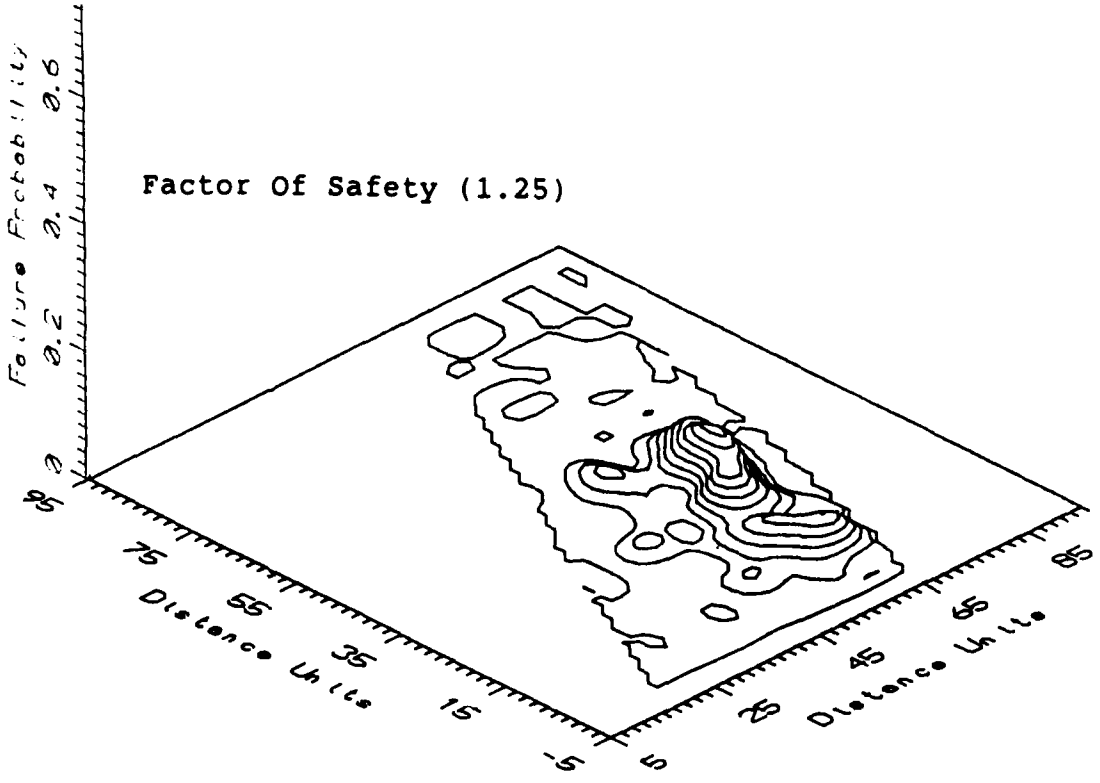


FIGURE 30. Combined Wing. Stochastically Best Wing vs. Factor of Safety Wing

Recommendations

Several recomendations were derived from this study:

(1) The entire scope could be expanded in a follow on study. The model doesn't have to be any larger, but additional element types, or more complex geometry, might provide additional insight to the utility of the method.

(2) Introduce additional levels of complexity into the random variables. For example, have more than one loading condition available, and allow the way elements are damaged to expand.

(3) Introduce additional random variables into the analysis. Let a variable representing fatigue cracking enter the picture, for example.

(4) Obtain a finite element code whose output could be more controlled than ASTROS. Its output always contained a lot of extraneous material, such as page headers, footers, and the like that had to be sifted through to get to the real data. That added to the programming complexity, as well as added time to each run.

(5) Try to find a higher speed computer. The computer used in this study was adequate for this work, but would not be fast enough in a significantly more complex project.

Appendix A

Program Listings

```
      REAL LOC,MEAN
      CHARACTER REST*64
      COMMON  INIT, ISEED
      INIT=-1
C
C     Note the input file is broken up into small files for easier handli
C
      OPEN (6,FILE='velmnt.dat',STATUS='OLD')
      OPEN (7,FILE='icw.inp',STATUS='OLD')
      OPEN (8,FILE='iseed', STATUS='OLD')
      OPEN (9,FILE='vforce.dat',STATUS='OLD')
      OPEN (10,FILE='control.dat',STATUS='OLD')
      OPEN (11,FILE='vgrid.dat',STATUS='OLD')
      OPEN (12,FILE='vprop.dat',STATUS='OLD')
      OPEN (14,FILE='broke.elem',STATUS='OLD')
C
      READ(8,1000)ISEED
C
5     READ(10,500,END=10) REST
      WRITE(7,500) REST
      GO TO 5
C
10    CALL FORCE
C
C     *URAND* returns uniform r.n. to localize damage in wing,
C     based approximately on percentages of areas normal to wing
C
15    LOC=URAND (ISEED)
      LOC=LOC*100
C
      IF (LOC.LT.90.0) CALL QUAD(ISEED)
      IF (90.0.LE.LOC.AND.LOC.LT.99.0) CALL SHEAR(ISEED)
      IF (99.0.LE.LOC.AND.ISEED.GT.4000) then
        CALL TRIANG(ISEED)
      elseif(99.0.le.loc.and.iseed.lt.4001) then
        go to 15
      ENDIF
C
C
1000  FORMAT(I7)
500   FORMAT(A64)
C
C
      STOP
      END
C
C
      SUBROUTINE TRIANG(ISEED)
      CHARACTER TYPE*8, POINTS*32, REST*64
      INTEGER ELEM, MAT, PROP
```

```
C
      WRITE(14,2000)1,2

C     Looks for triangular element and sub in new mat property
20       READ (6,1500,END=35) TYPE,ELEM,MAT,POINTS
         IF (LGE(TYPE,'CTRMEM')) THEN
           MAT=2
         ENDIF
           WRITE (7,1500) TYPE,ELEM,MAT,POINTS
      GO TO 20
C
C
   35 READ(12,500,END=40) REST
      WRITE(7,500) REST
      GO TO 35
C
40    READ(11,500,END=45)REST
      WRITE(7,500)REST
      GO TO 40
C
C
2000  FORMAT(2I5)
1500  FORMAT(A8,2I8,A32)
1000  FORMAT(I3,5X,I4)
500   FORMAT(A64)
C
C
45    RETURN
      END
C
C
      SUBROUTINE SHEAR(ISEED)
      CHARACTER TYPE*8, POINTS*32, REST*64
      INTEGER ELEM, MAT, SHRPNL, MATER, COUNTER
      TEMP=URAND(ISEED)
      SHRPNL=TEMP*55+65
C     Identifies one of 55 shear panels and adds 65 to it to
C
C
      IF(ISEED.LT.2001)CALL ROOT(SHRPNL)
      IF(ISEED.GT.2000 .AND. ISEED.LT.4001)CALL MID(SHRPNL)
      IF(ISEED.GT.4001)CALL TIP(SHRPNL)
C
C     Looks for shear panels in general and specifically for
C     shear panel matching location given by SHRPNL
C
      WRITE(14,2000)SHRPNL
C
40       READ (6,1500,end=50) TYPE,ELEM,MAT,POINTS
         IF(LGE(TYPE,'CSHEAR').AND.LLE(TYPE,'CSHEAR').AND.
```

```
     +      ELEM.EQ.SHRPNL) THEN
            MATER=MAT
            MAT=44
           ENDIF
           WRITE (7,1500) TYPE,ELEM,MAT,POINTS
       GO TO 40
C
C      Looks for PSHEAR card associated with damaged panel
50     READ (12,1500,END=55) TYPE,ELEM,MAT,POINTS
       IF(LGE(TYPE,'PSHEAR').AND.LLE(TYPE,'PSHEAR').AND.
     +    ELEM.EQ.MATER) THEN
           WRITE(7,1500)TYPE,44,3,POINTS
           ENDIF
       WRITE (7,1500) TYPE,ELEM,MAT,POINTS
       GO TO 50
C
55     READ(11,500,END=60)REST
       WRITE(7,500)REST
       GO TO 55
C
2000   FORMAT(I5)
1500   FORMAT (A8,2I8,A32)
1000   FORMAT (I3,5X,I4)
500    FORMAT (A64)
C
C
60     RETURN
       END
C
C
       SUBROUTINE QUAD(ISEED)
       CHARACTER TYPE*8, POINTS*32, REST*64
       INTEGER ELEM,MAT,MEMB1,MEMB2,NUMB
C      Sets up the two specific quadmems to be reset
57     TNUMB=URAND(ISEED)
       NUMB=TNUMB*30
       MEMB2=(NUMB+2)*2
       MEMB1=MEMB2-1
       IF(ISEED.LE.2000 .AND. MEMB1.GE.41) THEN
         GO TO 58
       ELSEIF(ISEED.GE.2001 .AND. ISEED.LE.4000 .AND. MEMB1.GE.25 .AND.
     +MEMB1.LE.40)THEN
         GO TO 58
       ELSEIF(ISEED.GE.4001 .AND. MEMB1.LE.23) THEN
         GO TO 58
       ENDIF
       GO TO 57
C      Looks for specific cquad member and the one below it.
C
58     WRITE(14,2000)MEMB1,MEMB2, ISEED
```

```
60     READ (6,1500,END=70) TYPE,ELEM,MAT,POINTS
       IF(LLE(TYPE,'CQDMEM1 ').AND.ELEM.EQ.MEMB1.OR.ELEM.EQ.MEMB2) THEN
C
          MATER=MAT
          MAT=43
C
       ENDIF
       WRITE (7,1500) TYPE,ELEM,MAT,POINTS
       GO TO 60
C      Reset their property cards
C
70     READ (12,1500,END=75) TYPE,ELEM,MAT,POINTS
       IF(LLE(TYPE,'PQDMEM1 ').AND.ELEM.EQ.MATER) THEN
         WRITE (7,1500) TYPE,43,3,POINTS
       ENDIF
       WRITE (7,1500) TYPE,ELEM,MAT,POINTS
       GO TO 70
C
75     READ(11,500,END=80) REST
       WRITE(7,500) REST
       GO TO 75
C
2000   FORMAT(2I5,I5)
1500   FORMAT(A8,2I8,A32)
1000   FORMAT(I3,5X,I4)
500    FORMAT(A64)
C
   80  RETURN
       END
C
C
       SUBROUTINE FORCE
C
       CHARACTER FRCE*32, VECTOR*24
       REAL MAG, MEAN
C
80       READ (9,1500,END=85) FRCE,MAG,VECTOR
         SDEV=0.1*MAG
         Y=GAUSS( MAG, SDEV )
         MAG=Y
         WRITE(7,1500) FRCE,MAG,VECTOR
         GO TO 80
 1500  FORMAT(A32,F8.2,A24)
C
C
85     RETURN
       END
C
C
       SUBROUTINE ROOT(SHRPNL)
```

```
      INTEGER SHRPNL

90    IF(SHRPNL.GE.89 .AND. SHRPNL.LE.96)RETURN
      IF(SHRPNL.GE.101 .AND. SHRPNL.LE.103)RETURN
      IF(SHRPNL.GE.109 .AND. SHRPNL.LE.111)RETURN
      IF(WHRPNL.GE.117 .AND. SHRPNL.LE.119)RETURN
      TEMP=URAND(ISEED)
      SHRPNL=TEMP*55+65
      GO TO 90
      END
C
      SUBROUTINE MID(SHRPNL)

      INTEGER SHRPNL

91    IF(SHRPNL.GE.78 .AND. SHRPNL.LE.88)RETURN
      IF(SHRPNA.GE.99 .AND. SHRPNL.LE.100)RETURN
      IF(SHRPNL.GE.107 .AND. SHRPNL.LE.108)RETURN
      IF(SHRPNL.GE.114 .AND. SHRPNL.LE.115)RETURN
      TEMP=URAND(ISEED)
      SHRPNL=TEMP*55+65
      GO TO 91
      END

      SUBROUTINE TIP(SHRPNL)

      INTEGER SHRPNL

92    IF(SHRPNL.GE.65 .AND. SHRPNL.LE.76)RETURN
      IF(SHRPNL.GE.97 .AND. SHRPNL.LE.98)RETURN
      IF(SHRPNL.GE.105 .AND. SHRPNL.LE.106)RETURN
      IF(SHRPNL.GE.113 .AND. SHRPNL.LE.114)RETURN
      TEMP=URAND(ISEED)
      SHRPNL=TEMP*55+64
      GO TO 92
      END
C
      REAL FUNCTION  GAUSS ( MEAN, STD )
C
C==GAUSS=======================================================
C
      REAL     MEAN
C      COMMON  INIT
C
C
      DATA ISET /0/
C
C    #INITIALIZATION OF =URAND= FOR THE RANDOM NUMBER SEQUENCE.
      IF  ( ISET .EQ. 0 )  THEN
```

```
C
C     #THE FOLLOWING COMPUTATIONS PRODUCE TWO RANDOM DEVIATES.  RETURN
C      ONE AND SAVE THE OTHER FOR THE NEXT CALL.  USE ODDCAL AS A CUE
C      FOR WHAT TO DO NEXT TIME IN.
C
C         #PICK TWO UNIFORM DEVIATES IN THE SQUARE EXTENDING
C          FROM -1 TO +1.
  100      V1 = 2.*URAND(ISEED) - 1.
           V2 = 2.*URAND(ISEED) - 1.
C
C         #SEE IF V1 AND V2 ARE IN THE UNIT CIRCLE.  IF NOT, TRY AGAIN.
           R = V1**2 + V2**2
           IF  ( R .GE. 1. )  GO TO  100
C
C         #MAKE THE BOX-MULLER TRANSFORMATION TO PRODUCE TWO NORMAL
C          DEVIATES WITH MEAN ZERO AND STANDARD DEVIATION ONE.
           TEMP = SQRT(-2.*LOG(R)/R)
           ENORM = V1*TEMP
           ONORM = V2*TEMP
           GAUSS = MEAN + STD*ONORM
           ISET = 1
       ELSE
C
C         #USE ENORM FROM PREVIOUS CALL TO COMPUTE GAUSS.
C
           GAUSS = MEAN + STD*ENORM
           ISET = 0
       ENDIF
C
       RETURN
       END
       REAL FUNCTION  URAND (ISEED)
C
C=URAND=========================================================
       COMMON  INIT
C2================================================================
C
       INTEGER  A1 ,            A2 ,            A3
       INTEGER  C1 ,            C2 ,            C3
       INTEGER  M1 ,            M2 ,            M3
       PARAMETER  (M1 = 259200, A1 = 7141, C1 = 54773)
       PARAMETER  (M2 = 134456, A2 = 8121, C2 = 28411)
       PARAMETER  (M3 = 243000, A3 = 4561, C3 = 51349)
       PARAMETER  (NSHUF = 30)
       REAL     UDEV(NSHUF)
C
       RM1 = 1./M1
       RM2 = 1./M2
C
C
```

```
C     #INITIALIZE ON FIRST CALL.
      IF  ( INIT .LT. 0 )  THEN
C
C          OPEN(8,FILE='iseed',STATUS='OLD')
           REWIND(8)
           READ(8,115) ISEED
           REWIND(8)
           ISEED=10+ISEED
           WRITE(8,115) ISEED
 115       FORMAT(I7)
C
           I1 = MOD(ABS(C1-ISEED), M1)
           I1 = MOD(A1*I1+C1, M1)
           I2 = MOD(I1, M2)
           I1 = MOD(A1*I1+C1, M1)
           I3 = MOD(I1, M3)
C
C         #INITIALIZE THE SHUFFLING VECTOR FOR RANDOM SEQUENCE.
           DO  11 I = 1, NSHUF
               I1 = MOD(A1*I1+C1, M1)
               I2 = MOD(A2*I2+C2, M2)
               UDEV(I) = (REAL(I1) + REAL(I2)*RM2)*RM1
  11       CONTINUE
           INIT = 1
      ENDIF
C
C     #GENERATE THE NEXT NUMBER IN EACH INTEGER SEQUENCE I1, I2, I3.
      I1 = MOD(A1*I1+C1, M1)
      I2 = MOD(A2*I2+C2, M2)
      I3 = MOD(A3*I3+C3, M3)
C
C     #USE THE THIRD INTEGER SEQUENCE TO GET A NUMBER BETWEEN
C      1 AND NSHUF.
      I = 1 + (NSHUF*I3) / M3
C
C     #RETURN THAT VECTOR ENTRY.
      URAND = UDEV(I)
C
C     #REFILL SHUFFLING VECTOR WHERE SAMPLE HAS BEEN USED.
      UDEV(I) = (REAL(I1) + REAL(I2)*RM2)*RM1
C
      RETURN
      END
```

```
      CHARACTER INPUT*80, DATA*48, TARGET*10, BUSTED*10
      INTEGER J, N, LOOP, POINT
      REAL STRESS(158),DISP(88),PRESS1,PRESS2,MAXSTR
      REAL MAXD1,MAXD3,MAXD5,MAXD7,MAXD9
      DATA J/1/
      DATA N/0/
      DATA NSET/0/
      DATA LOOP/0/
      DATA POINT/0/
      DATA LOOPD/0/
      PARAMETER (MAXSTR=55750.0)
      PARAMETER (MAXD1=12.0, MAXD3=12.75, MAXD5=13.5)
      PARAMETER (MAXD7=14.25, MAXD9=15.0)
      PARAMETER (NNODE=88)
      OPEN(10,FILE='icw.out', STATUS='OLD')
      OPEN(11,FILE='stress.dat',STATUS='OLD')
      OPEN(12,FILE='displ.dat',STATUS='OLD')
      OPEN(14,FILE='iseed',STATUS='OLD')
      OPEN(15,FILE='damage',STATUS='OLD')
      OPEN(20,FILE='broke.elem',STATUS='OLD')
C
      READ (14,1000)ISEED
      READ (20,1200)BUSTED
C
C
100   READ (10,2000)INPUT
C
C     Looks for node heading on displacement page
      POINT=INDEX(INPUT,'POINT ID.')
200   IF(POINT.GT.0)THEN
        GO TO 225
       ELSE
        GO TO 100
      ENDIF
C
225   LOOPD=LOOPD+1
      GO TO (300,300)LOOPD
C
250   READ (10,2000)INPUT
C     Looks for element heading on stress pages
      POINT=INDEX(INPUT,'ELEMENT')
      IF(POINT.GT.0) THEN
        GO TO 275
        ELSE
        GO TO 250
        ENDIF
275   LOOP=LOOP+1
C
C     Recycles to look for various elemental stress pages
      GO TO (400,600,800,900) LOOP
```

```
C
C     Computes displacement magnitude for each element and writes to file
C
300   READ(10,3000,ERR=100) J,DISPL1,DISPL2,DISPL3
      DISP(J)=SQRT(DISPL1**2+DISPL2**2+DISPL3**2)
      IF(J.EQ.NNODE)THEN
        GO TO 250
        ELSE
        GO TO 300
      ENDIF
C
C     This loop reads and processes the quad member stresses
C     Following acts as line feed to get to lines of data
400   READ(10,2000) INPUT
C     Following ERR statement jumps to second page of quad stresses
500   READ(10,5000,ERR=250) M,PRESS1,PRESS2
C     Following computes Mises-Hinkley Stress Criterion, writes it to file
      STRESS(M)=SQRT(PRESS1**2+PRESS2**2-PRESS1*PRESS2)
      IF(M.EQ.64)THEN
      GO TO 250
      else
      GO TO 500
      endif
C
C     Following acts as a line feed
C
600   READ(10,2000)INPUT
C     Following loop reads rod stresses
625   READ(10,6000) M, STRESS(M), STRESS(M+1)
C
      IF (M.LT.156) THEN
        GO TO 625
C
      ELSE
        READ(10,8000) M,STRESS(M)
        GO TO 250
C
      ENDIF
C
800   READ(10,2000) INPUT
C
C     Following reads shear panel stresses
C
825   READ(10,7000) M,STRESS(M),STRESS(M+1)
      IF(M.LT.117) THEN
        GO TO 825
      ELSE
        READ(10,7500) M,STRESS(M)
        GO TO 250
C
```

```
      ENDIF
C
      GO TO 825
C
C     Following reads triangular panel stresses
C
900   READ(10,2000) INPUT
C
925   READ(10,5000,err=10) M,PRESS1,PRESS2
C
      STRESS(M)=SQRT(PRESS1**2+PRESS2**2-PRESS1*PRESS2)
      IF(M.eq.2)go to 10
C
      GO TO 925
C
C     Following puts pointers at end of cumulative data files
C     and writes new data lines.
C
10    READ(12,2000)INPUT
C
c     GO TO 10
C
c0    WRITE(12,4000) (DISP(K), K=1,88)
C
C
30    READ(11,2000,END=40) INPUT
C
      GO TO 30
C
40    WRITE(11,9000) (STRESS(I), I=1,158)
C
50    READ(15,2000,END=55)INPUT
      GO TO 50
C
55    DO 60 I=1,158
        IF(STRESS(I).GT.MAXSTR)THEN
          NSET = 1
      ENDIF
60    CONTINUE
C
      IF(NSET.EQ.1) then
      go to 62
      else
      WRITE(15,2500)ISEED,BUSTED
      ENDIF
C
62    DO 65 I=1,158
        IF(STRESS(I).GT.MAXSTR)THEN
          WRITE(15,1500)ISEED,BUSTED,I,STRESS(I)
      ENDIF
```

```
65    CONTINUE
c     if(disp(1).gt.maxd1) write(15,1600)iseed,1,disp(1)
c     if(disp(3).gt.maxd3) write(15,1600)iseed,3,disp(3)
c     if(disp(5).gt.maxd5) write(15,1600)iseed,5,disp(5)
c     if(disp(7) gt.maxd7) write(15,1600)iseed,7,disp(7)
c     if(disp(9).gt.maxd9) write(15,1600)iseed,9,disp(9)
c     DO 70 I=1,9,2
c       IF(DISP(I).GT.MAXDIS(I))THEN
c         WRITE(15,1600)ISEED,I,DISP(I)
c       ENDIF
c0    CONTINUE
C
1000  FORMAT(I7)
1200  FORMAT(A10)
1500  FORMAT('RUN',I5,' BRKN ELEMS',A10,'  FLD ELEM',I5,' STR=',E14.6)
1600  FORMAT('RUN # ',I5,'  NODE # ',I5,'  DISPLACEMENT= ',E14.6)
2000  FORMAT(A80)
2500  FORMAT('Run',I5, ' Broken Elements: ',A10,'  No Failures')
3000  FORMAT(11X,I7,15X,3E14.6)
4000  FORMAT(88E14.6)
5000  FORMAT(I9,64X,2E14.6)
6000  FORMAT(I13,E14.6,45X,E14.6)
7000  FORMAT(I14,E16.6,45X,E16.6)
7500  FORMAT(I14,E16.6)
8000  FORMAT(I13,E14.6)
9000  FORMAT(158E14.6)
C
      STOP
      END
```

```
      real stress(6000,158),sdev(158),mean(158),sum(158)
      real sum2(158),prob(158)

C     Loops is the total number of iterations you've done w/ Astros
C     Remember to update this number to reflect the total number of runs
C     to be evaluated

      parameter (loops = 200,sigmax=52500)
      parameter (c1=.196854,c2=.115194,c3=.000344,c4=.019527)

      open(10,file='stress',status='old')
      open(20,file='output',status='old')
      open(30,file='prob.dat',status='old')

      do 100 i=1,loops

      read(10,1000,end=100)(stress(i,j),j=1,158)

100   continue

      do 300 j=1,158
      do 200 i=1,loops

      sum(j)=stress(i,j)+sum(j)
      sum2(j)=stress(i,j)**2+sum2(j)

200   continue
300   continue

      do 400 j=1,158

      mean(j)=sum(j)/loops
      sdev(j)=sqrt(sum2(j)/loops-mean(j)**2)


400   continue

      do 500 j=1,158

      write(20,2000)j,mean(j),sdev(j)

500   continue

      do 600 m=1,158
      x=(sigmax-mean(m))/sdev(m)
      prob(m)=1-.5*(1+c1*x+c2*x**2+c3*x**3+c4*x**4)**-4
      write(30,3000)m,prob(m)

600   continue


1000  format(158E14.6)
2000  format('mean in element',I4,'  is',E14.6,' std dev is',E14.6)
3000  format(i5,f8.3)

      stop
      end
```

```
c     This program transforms a list of probabilties
c     into an input file, coordinate vs. probabilty,
c     for input into a 3-D graphics program. additional
c     points are generated to give the graph a smoother surface.

      character misc*64
      real x(88),y(88),xbar(1000),ybar(1000),prob(158)

      open(10,file='vgrid.dat',status='old')
      open(20,file='velmnt.dat',status='old')
      open(30,file='prob.dat',status='old')
      open(40,file='wing.grid',status='old')
c     Gets past first five lines to get to the element
c     and its corresponding node numbers.

      do 5 m=1,5
      read(10,5000)misc
5     continue
c     reads in element and its node numbers
      do 10 m=1,88
      read(10,1000)n,x(n),y(n)
10    continue

      do 15 m=1,119

        read(30,3000)prob(m)

15    continue
c     finds center of triangular element
      read(20,2000)n,i,j,k,l
      xbar(n)=(x(i)+x(j)+x(k))/3
      ybar(n)=(y(i)+y(j)+y(k))/3

      read(20,5000)misc
c     following loop locates four corners of quad elements
c     and assigns coordinates to them
      do 20 m=3,63,2

        read(20,2000)n,i,j,k,l

         ex= abs(x(i)-x(j))
         wy= abs(y(i)-y(k))

         xbar(m)=(x(i)+x(j)+x(k)+x(l))/4
         xbar(m+100)= x(i)+.1*ex
         xbar(m+200)= x(j)-.1*ex
         xbar(m+300)= x(k)-.1*ex
         xbar(m+400)= x(l)+.1*ex

         ybar(m)=(y(i)+y(j)+y(k)+y(l))/4
```

```
          ybar(m+100)= y(i)-.1*wy
          ybar(m+200)= y(j)-.1*wy
          ybar(m+300)= y(k)+.1*wy
          ybar(m+400)= y(l)+.1*wy

        read(20,5000) misc

20      continue
c       following loop locates the line of each shear element
c       and gives coordinates to several points.  figures out
c       which direction the element is pointing to make sure
c       the coordinates follow the line correctly.
        do 30 m=1,55

        read(20,2000)n,i,j,k,l

        ex= abs(x(i)-x(k))
        wy= abs(y(i)-y(k))

        xbar(m+500)=x(i)+.01*x(i)
        xbar(m+600)=x(k)+.01*x(k)
        xbar(m+700)=(x(i)+x(k))/2

        if (x(i).gt.x(k)) then

        xbar(m+800)=x(k)+.25*ex
        xbar(m+900)=x(k)+.75*ex

        else

        xbar(m+800)=x(i)+.25*ex
        xbar(m+900)=x(i)+.75*ex

        endif

        ybar(m+500)=y(i)+.01*y(i)
        ybar(m+600)=y(k)+.01*y(k)
        ybar(m+700)=(y(i)+y(k))/2

        if (y(i).gt.y(k)) then

        ybar(m+800)=y(k)+.75*wy
        ybar(m+900)=y(k)+.25*wy

        else

        ybar(m+800)=y(i)+.25*wy
        ybar(m+900)=y(i)+.75*wy

        endif
```

```
30    continue
c     writes each coordinate and its associated prob to file
      write(40,4000)xbar(1),ybar(1),prob(1)

      do 400 m=3,63,2

          do 450 n=0,400,100

            write(40,4000) xbar(m+n), ybar(m+n), prob(m)

450       continue

400   continue

      do 550 m=1,55
      do 500 n=500,900,100
      write(40,4000)xbar(m+n),ybar(m+n),prob(m+64)
500   continue
550   continue


1000  format(8x,i8,8x,2f8.3)
2000  format(8x,i8,8x,4i8)
3000  format(5x,f8.3)
4000  format(f8.3,f8.3,f8.3)
5000  format(a64)

      stop
      end
```

Feb 23 19:55 1989 post3.for Page 1

```
c       This program provides a means to use pre.for,
c       ASTROS, post1.for, run.x, and post2.for in an
c       optimizing scheme, with optimize.x as the
c       controller.  Note this program does not allow
c       an element to be reduced in weight.

        character type*8

        open(10,file='prob.dat',status='old')
        open(20,file='vprop.dat',status='old')
        open(30,file='vp.dat',status='old')
        open(40,file='stress.dat',status='old')
        open(50,file='iseed',status='old')

10      read(20,1000,end=100)type,n,mat,thick
        read(10,2000)num,prob
        x=abs(1.0-prob)
        if(x.ge.0.005)thick=thick+0.005
        write(30,3000)type,n,mat,thick
        go to 10

100     write(40,4000)
        write(50,5000)1

1000    format(a8,2i8,f8.4)
2000    format(i5,f8.3)
3000    format(a8,2i8,f8.4)
4000    format('    ')
5000    format(i5)
        end
```

Feb 25 00:16 1989 run.x

```
#
set n=0
while ($n != 400)
  pre.x
  astros icw.inp
  post1.x
  @ n++
end
```

Feb 25 00:16 1989 optimize.x

```
#
set m=0
while ($m != 5)
  run.x
  p600.x
  post3.x
  cp vp.dat vprop.dat
  @ m++
end
```

Appendix B

Sample ASTROS Input

```
ASSIGN DATABASE ICW TEST NEW DELETE
SOLUTION
TITLE=      INTERMEDIATE COMPLEXITY WING
ANALYZE
   BOUNDARY SPC = 1
     STATICS (MECH = 1)
        PRINT DISP=ALL, STRESS=ALL
END
BEGIN BULK
FORCE           1       3          15.17  0.0000  0.0000  1.0000
FORCE           1       4          15.26  0.0000  0.0000  1.0000
FORCE           1       5       -1538.43  1.0000  0.0000  0.0000
FORCE           1       5       -3924.44  0.0000  1.0000  0.0000
FORCE           1       5         570.04  0.0000  0.0000  1.0000
FORCE           1       6        1393.27  1.0000  0.0000  0.0000
FORCE           1       6        4481.66  0.0000  1.0000  0.0000
FORCE           1       6         618.14  0.0000  0.0000  1.0000
FORCE           1       7          48.87  0.0000  0.0000  1.0000
FORCE           1       8          50.47  0.0000  0.0000  1.0000
FORCE           1       9       -5165.61  1.0000  0.0000  0.0000
FORCE           1       9       -4908.70  0.0000  1.0000  0.0000
FORCE           1       9         490.31  0.0000  0.0000  1.0000
FORCE           1      10        4636.29  1.0000  0.0000  0.0000
FORCE           1      10        5067.86  0.0000  1.0000  0.0000
FORCE           1      10         566.79  0.0000  0.0000  1.0000
FORCE           1       1         106.29  1.0000  0.0000  0.0000
FORCE           1       1       -4164.07  0.0000  1.0000  0.0000
FORCE           1       1         489.42  0.0000  0.0000  1.0000
FORCE           1       2        -120.12  1.0000  0.0000  0.0000
FORCE           1       2        3445.47  0.0000  1.0000  0.0000
FORCE           1       2         424.33  0.0000  0.0000  1.0000
FORCE           1      11          86.57  0.0000  0.0000  1.0000
FORCE           1      12         101.77  0.0000  0.0000  1.0000
FORCE           1      13         131.35  0.0000  0.0000  1.0000
FORCE           1      14         133.78  0.0000  0.0000  1.0000
FORCE           1      15         149.66  0.0000  0.0000  1.0000
FORCE           1      16         119.26  0.0000  0.0000  1.0000
FORCE           1      17       -3099.10  1.0000  0.0000  0.0000
FORCE           1      17        1034.29  0.0000  1.0000  0.0000
FORCE           1      17         532.08  0.0000  0.0000  1.0000
FORCE           1      18        2693.56  1.0000  0.0000  0.0000
FORCE           1      18       -1333.79  0.0000  1.0000  0.0000
FORCE           1      18         510.58  0.0000  0.0000  1.0000
FORCE           1      19        1224.43  1.0000  0.0000  0.0000
FORCE           1      19        -457.83  0.0000  1.0000  0.0000
FORCE           1      19         414.60  0.0000  0.0000  1.0000
FORCE           1      20       -1367.42  1.0000  0.0000  0.0000
FORCE           1      20         527.02  0.0000  1.0000  0.0000
FORCE           1      20         343.27  0.0000  0.0000  1.0000
FORCE           1      21         169.08  0.0000  0.0000  1.0000
```

FORCE	1	22	177.12	0.0000	0.0000	1.0000
FORCE	1	23	172.57	0.0000	0.0000	1.0000
FORCE	1	24	149.96	0.0000	0.0000	1.0000
FORCE	1	25	181.42	0.0000	0.0000	1.0000
FORCE	1	26	200.56	0.0000	0.0000	1.0000
FORCE	1	27	-2252.94	1.0000	0.0000	0.0000
FORCE	1	27	963.52	0.0000	1.0000	0.0000
FORCE	1	27	501.60	0.0000	0.0000	1.0000
FORCE	1	28	1665.66	1.0000	0.0000	0.0000
FORCE	1	28	-864.64	0.0000	1.0000	0.0000
FORCE	1	28	420.19	0.0000	0.0000	1.0000
FORCE	1	29	833.74	1.0000	0.0000	0.0000
FORCE	1	29	-330.92	0.0000	1.0000	0.0000
FORCE	1	29	323.85	0.0000	0.0000	1.0000
FORCE	1	30	-893.03	1.0000	0.0000	0.0000
FORCE	1	30	394.44	0.0000	1.0000	0.0000
FORCE	1	30	290.28	0.0000	0.0000	1.0000
FORCE	1	31	181.44	0.0000	0.0000	1.0000
FORCE	1	32	161.47	0.0000	0.0000	1.0000
FORCE	1	33	185.69	0.0000	0.0000	1.0000
FORCE	1	34	207.31	0.0000	0.0000	1.0000
FORCE	1	35	198.26	0.0000	0.0000	1.0000
FORCE	1	36	172.47	0.0000	0.0000	1.0000
FORCE	1	37	-2347.27	1.0000	0.0000	0.0000
FORCE	1	37	893.76	0.0000	1.0000	0.0000
FORCE	1	37	561.65	0.0000	0.0000	1.0000
FORCE	1	38	2261.60	1.0000	0.0000	0.0000
FORCE	1	38	-941.98	0.0000	1.0000	0.0000
FORCE	1	38	479.01	0.0000	0.0000	1.0000
FORCE	1	39	945.21	1.0000	0.0000	0.0000
FORCE	1	39	-421.50	0.0000	1.0000	0.0000
FORCE	1	39	390.19	0.0000	0.0000	1.0000
FORCE	1	40	-778.56	1.0000	0.0000	0.0000
FORCE	1	40	353.86	0.0000	1.0000	0.0000
FORCE	1	40	347.96	0.0000	0.0000	1.0000
FORCE	1	41	206.87	0.0000	0.0000	1.0000
FORCE	1	42	181.70	0.0000	0.0000	1.0000
FORCE	1	43	189.96	0.0000	0.0000	1.0000
FORCE	1	44	237.12	0.0000	0.0000	1.0000
FORCE	1	45	200.79	0.0000	0.0000	1.0000
FORCE	1	46	206.23	0.0000	0.0000	1.0000
FORCE	1	47	-2182.20	1.0000	0.0000	0.0000
FORCE	1	47	968.95	0.0000	1.0000	0.0000
FORCE	1	47	536.20	0.0000	0.0000	1.0000
FORCE	1	48	2296.42	1.0000	0.0000	0.0000
FORCE	1	48	-946.42	0.0000	1.0000	0.0000
FORCE	1	48	494.70	0.0000	0.0000	1.0000
FORCE	1	49	1103.36	1.0000	0.0000	0.0000
FORCE	1	49	-452.79	0.0000	1.0000	0.0000
FORCE	1	49	417.00	0.0000	0.0000	1.0000

FORCE	1	50	-912.25	1.0000	0.0000	0.0000
FORCE	1	50	462.44	0.0000	1.0000	0.0000
FORCE	1	50	435.68	0.0000	0.0000	1.0000
FORCE	1	51	234.88	0.0000	0.0000	1.0000
FORCE	1	52	194.34	0.0000	0.0000	1.0000
FORCE	1	53	167.27	0.0000	0.0000	1.0000
FORCE	1	54	204.93	0.0000	0.0000	1.0000
FORCE	1	55	223.26	0.0000	0.0000	1.0000
FORCE	1	56	232.96	0.0000	0.0000	1.0000
FORCE	1	57	-2246.54	1.0000	0.0000	0.0000
FORCE	1	57	956.33	0.0000	1.0000	0.0000
FORCE	1	57	612.16	0.0000	0.0000	1.0000
FORCE	1	58	2202.02	1.0000	0.0000	0.0000
FORCE	1	58	-989.04	0.0000	1.0000	0.0000
FORCE	1	58	625.45	0.0000	0.0000	1.0000
FORCE	1	59	1314.61	1.0000	0.0000	0.0000
FORCE	1	59	-494.60	0.0000	1.0000	0.0000
FORCE	1	59	515.95	0.0000	0.0000	1.0000
FORCE	1	60	-1233.54	1.0000	0.0000	0.0000
FORCE	1	60	502.27	0.0000	1.0000	0.0000
FORCE	1	60	312.37	0.0000	0.0000	1.0000
FORCE	1	61	230.44	0.0000	0.0000	1.0000
FORCE	1	62	211.40	0.0000	0.0000	1.0000
FORCE	1	63	212.10	0.0000	0.0000	1.0000
FORCE	1	64	184.19	0.0000	0.0000	1.0000
FORCE	1	65	208.58	0.0000	0.0000	1.0000
FORCE	1	66	195.87	0.0000	0.0000	1.0000
FORCE	1	67	-1676.00	1.0000	0.0000	0.0000
FORCE	1	67	745.21	0.0000	1.0000	0.0000
FORCE	1	67	383.15	0.0000	0.0000	1.0000
FORCE	1	68	1418.60	1.0000	0.0000	0.0000
FORCE	1	68	-507.59	0.0000	1.0000	0.0000
FORCE	1	68	501.30	0.0000	0.0000	1.0000
FORCE	1	69	1696.74	1.0000	0.0000	0.0000
FORCE	1	69	-232.75	0.0000	1.0000	0.0000
FORCE	1	69	553.35	0.0000	0.0000	1.0000
FORCE	1	70	-1440.36	1.0000	0.0000	0.0000
FORCE	1	70	246.59	0.0000	1.0000	0.0000
FORCE	1	70	577.07	0.0000	0.0000	1.0000
FORCE	1	71	240.01	0.0000	0.0000	1.0000
FORCE	1	72	186.24	0.0000	0.0000	1.0000
FORCE	1	73	208.65	0.0000	0.0000	1.0000
FORCE	1	74	160.27	0.0000	0.0000	1.0000
FORCE	1	75	180.50	0.0000	0.0000	1.0000
FORCE	1	76	153.32	0.0000	0.0000	1.0000
FORCE	1	77	-670.41	1.0000	0.0000	0.0000
FORCE	1	77	145.01	0.0000	1.0000	0.0000
FORCE	1	77	232.81	0.0000	0.0000	1.0000
FORCE	1	78	795.53	1.0000	0.0000	0.0000
FORCE	1	78	-133.64	0.0000	1.0000	0.0000

FORCE	1	78		240.06	0.0000	0.0000	1.0000
CTRMEM	1	1	1	3	11		
CTRMEM	2	1	2	4	12		
CQDMEM1	3	3	3	5	13	11	
CQDMEM1	4	3	4	6	14	12	
CQDMEM1	5	3	5	7	15	13	
CQDMEM1	6	3	6	8	16	14	
CQDMEM1	7	4	7	9	17	15	
CQDMEM1	8	4	8	10	18	16	
CQDMEM1	9	3	1	11	21	19	
CQDMEM1	10	3	2	12	22	20	
CQDMEM1	11	3	11	13	23	21	
CQDMEM1	12	3	12	14	24	22	
CQDMEM1	13	5	13	15	25	23	
CQDMEM1	14	5	14	16	26	24	
CQDMEM1	15	5	15	17	27	25	
CQDMEM1	16	5	16	18	28	26	
CQDMEM1	17	5	19	21	31	29	
CQDMEM1	18	5	20	22	32	30	
CQDMEM1	19	6	21	23	33	31	
CQDMEM1	20	6	22	24	34	32	
CQDMEM1	21	6	23	25	35	33	
CQDMEM1	22	6	24	26	36	34	
CQDMEM1	23	7	25	27	37	35	
CQDMEM1	24	7	26	28	38	36	
CQDMEM1	25	4	29	31	41	39	
CQDMEM1	26	4	30	32	42	40	
CQDMEM1	27	8	31	33	43	41	
CQDMEM1	28	8	32	34	44	42	
CQDMEM1	29	8	33	35	45	43	
CQDMEM1	30	8	34	36	46	44	
CQDMEM1	31	8	35	37	47	45	
CQDMEM1	32	8	36	38	48	46	
CQDMEM1	33	6	39	41	51	49	
CQDMEM1	34	6	40	42	52	50	
CQDMEM1	35	9	41	43	53	51	
CQDMEM1	36	9	42	44	54	52	
CQDMEM1	37	10	43	45	55	53	
CQDMEM1	38	10	44	46	56	54	
CQDMEM1	39	11	45	47	57	55	
CQDMEM1	40	11	46	48	58	56	
CQDMEM1	41	7	49	51	61	59	
CQDMEM1	42	7	50	52	62	60	
CQDMEM1	43	10	51	53	63	61	
CQDMEM1	44	10	52	54	64	62	
CQDMEM1	45	12	53	55	65	63	
CQDMEM1	46	12	54	56	66	64	
CQDMEM1	47	13	55	57	67	65	
CQDMEM1	48	13	56	58	68	66	
CQDMEM1	49	7	59	61	71	69	

CQDMEM1	50	7	60	62	72	70
CQDMEM1	51	11	61	63	73	71
CQDMEM1	52	11	62	64	74	72
CQDMEM1	53	14	63	65	75	73
CQDMEM1	54	14	64	66	76	74
CQDMEM1	55	15	65	67	77	75
CQDMEM1	56	15	66	68	78	76
CQDMEM1	57	6	69	71	81	79
CQDMEM1	58	6	70	72	82	80
CQDMEM1	59	16	71	73	83	81
CQDMEM1	60	16	72	74	84	82
CQDMEM1	61	17	73	75	85	83
CQDMEM1	62	17	74	76	86	84
CQDMEM1	63	18	75	77	87	85
CQDMEM1	64	18	76	78	88	86
CSHEAR	65	20	1	2	4	3
CSHEAR	66	20	3	4	6	5
CSHEAR	67	20	5	6	8	7
CSHEAR	68	20	7	8	10	9
CSHEAR	69	20	1	2	12	11
CSHEAR	70	20	11	12	14	13
CSHEAR	71	20	13	14	16	15
CSHEAR	72	20	15	16	18	17
CSHEAR	73	20	19	20	22	21
CSHEAR	74	20	21	22	24	23
CSHEAR	75	20	23	24	26	25
CSHEAR	76	20	25	26	28	27
CSHEAR	77	20	29	30	32	31
CSHEAR	78	20	31	32	34	33
CSHEAR	79	20	33	34	36	35
CSHEAR	80	20	35	36	38	37
CSHEAR	81	20	39	40	42	41
CSHEAR	82	20	41	42	44	43
CSHEAR	83	20	43	44	46	45
CSHEAR	84	20	45	46	48	47
CSHEAR	85	20	49	50	52	51
CSHEAR	86	20	51	52	54	53
CSHEAR	87	20	53	54	56	55
CSHEAR	88	20	55	56	58	57
CSHEAR	89	21	59	60	62	61
CSHEAR	90	22	61	62	64	63
CSHEAR	91	20	63	64	66	65
CSHEAR	92	20	65	66	68	67
CSHEAR	93	23	69	70	72	71
CSHEAR	94	24	71	72	74	73
CSHEAR	95	20	73	74	76	75
CSHEAR	96	20	75	76	78	77
CSHEAR	97	20	1	2	20	19
CSHEAR	98	25	19	20	30	29
CSHEAR	99	26	29	30	40	39

CSHEAR	100	27	39	40	50	49
CSHEAR	101	28	49	50	60	59
CSHEAR	102	29	59	60	70	69
CSHEAR	103	30	69	70	80	79
CSHEAR	104	31	5	6	14	13
CSHEAR	105	32	13	14	24	23
CSHEAR	106	33	23	24	34	33
CSHEAR	107	34	33	34	44	43
CSHEAR	108	35	43	44	54	53
CSHEAR	109	36	53	54	64	63
CSHEAR	110	37	63	64	74	73
CSHEAR	111	38	73	74	84	83
CSHEAR	112	20	9	10	18	17
CSHEAR	113	44	17	18	28	27
CSHEAR	114	39	27	28	38	37
CSHEAR	115	40	37	38	48	47
CSHEAR	116	35	47	48	58	57
CSHEAR	117	36	57	58	68	67
CSHEAR	118	41	67	68	78	77
CSHEAR	119	42	77	78	88	87
CROD	120	19	1	2		
CROD	121	19	3	4		
CROD	122	19	5	6		
CROD	123	19	7	8		
CROD	124	19	9	10		
CROD	125	19	11	12		
CROD	126	19	13	14		
CROD	127	19	15	16		
CROD	128	19	17	18		
CROD	129	19	19	20		
CROD	130	19	21	22		
CROD	131	19	23	24		
CROD	132	19	25	26		
CROD	133	19	27	28		
CROD	134	19	29	30		
CROD	135	19	31	32		
CROD	136	19	33	34		
CROD	137	19	35	36		
CROD	138	19	37	38		
CROD	139	19	39	40		
CROD	140	19	41	42		
CROD	141	19	43	44		
CROD	142	19	45	46		
CROD	143	19	47	48		
CROD	144	19	49	50		
CROD	145	19	51	52		
CROD	146	19	53	54		
CROD	147	19	55	56		
CROD	148	19	57	58		
CROD	149	19	59	60		

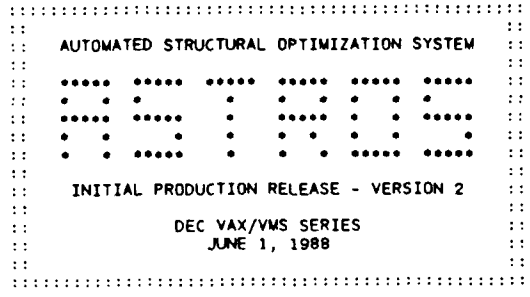
CROD	150	19	61	62
CROD	151	19	63	64
CROD	152	19	65	66
CROD	153	19	67	68
CROD	154	19	69	70
CROD	155	19	71	72
CROD	156	19	73	74
CROD	157	19	75	76
CROD	158	19	77	78
PTRMEM	1	1	.0312	
PTRMEM	2	3	.0312	
PQDMEM1	3	2	.0312	
PQDMEM1	4	2	.0416	
PQDMEM1	5	2	.0364	
PQDMEM1	6	2	.0624	
PQDMEM1	7	2	.0520	
PQDMEM1	8	2	.0884	
PQDMEM1	9	2	.1040	
PQDMEM1	10	2	.1040	
PQDMEM1	11	2	.1144	
PQDMEM1	12	2	.1875	
PQDMEM1	13	2	.1040	
PQDMEM1	14	2	.1144	
PQDMEM1	15	2	.1563	
PQDMEM1	16	2	.0884	
PQDMEM1	17	2	.1248	
PQDMEM1	18	2	.2031	
PROD	19	1	.0200	
PSHEAR	20	1	.0190	
PSHEAR	21	1	.0220	
PSHEAR	22	1	.0210	
PSHEAR	23	1	.0250	
PSHEAR	24	1	.0240	
PSHEAR	44	3	.0380	
PSHEAR	25	1	.0380	
PSHEAR	26	1	.0420	
PSHEAR	27	1	.0480	
PSHEAR	28	1	.0470	
PSHEAR	29	1	.0390	
PSHEAR	30	1	.0310	
PSHEAR	31	1	.0280	
PSHEAR	32	1	.0370	
PSHEAR	33	1	.0500	
PSHEAR	34	1	.0580	
PSHEAR	35	1	.0650	
PSHEAR	36	1	.0790	
PSHEAR	37	1	.1010	
PSHEAR	38	1	.1260	
PSHEAR	39	1	.0440	
PSHEAR	40	1	.0530	

```
PSHEAR          41        1    .0920
PSHEAR          42        1    .1000
GRDSET                                                                  456
MAT1            1 10.5E+06                      .300 .000259
MAT1            2 10.5E+06                      .300 .000259
MAT1            3 10.5E+04                      .300 .000259
SPC1     1        123      79       THRU     88
GRID             1           63.500   90.000    1.125
GRID             2           63.500   90.000   -1.125
GRID             3           70.833   90.000    1.313
GRID             4           70.833   90.000   -1.313
GRID             5           78.167   90.000    1.500
GRID             6           78.167   90.000   -1.500
GRID             7           85.500   90.000    1.313
GRID             8           85.500   90.000   -1.313
GRID             9           92.833   90.000    1.125
GRID            10           92.833   90.000   -1.125
GRID            11           69.686   87.471    1.349
GRID            12           69.686   87.471   -1.349
GRID            13           76.097   84.851    1.586
GRID            14           76.097   84.851   -1.586
GRID            15           82.746   82.133    1.427
GRID            16           82.746   82.133   -1.427
GRID            17           89.647   79.312    1.259
GRID            18           89.647   79.312   -1.259
GRID            19           57.266   77.669    1.279
GRID            20           57.266   77.669   -1.279
GRID            21           63.992   74.920    1.532
GRID            22           63.992   74.920   -1.532
GRID            23           70.962   72.071    1.799
GRID            24           70.962   72.071   -1.799
GRID            25           78.191   69.116    1.617
GRID            26           78.191   69.116   -1.617
GRID            27           85.692   66.050    1.424
GRID            28           85.692   66.050   -1.424
GRID            29           51.032   65.339    1.433
GRID            30           51.032   65.339   -1.433
GRID            31           58.297   62.369    1.715
GRID            32           58.297   62.369   -1.715
GRID            33           65.826   59.291    2.012
GRID            34           65.826   59.291   -2.012
GRID            35           73.635   56.100    1.807
GRID            36           73.635   56.100   -1.807
GRID            37           81.738   52.787    1.590
GRID            38           81.738   52.787   -1.590
GRID            39           44.799   53.008    1.587
GRID            40           44.799   53.008   -1.587
GRID            41           52.603   49.818    1.898
GRID            42           52.603   49.818   -1.898
GRID            43           60.691   46.512    2.225
```

```
GRID            44          60.691   46.512   -2.225
GRID            45          69.079   43.083    1.997
GRID            46          69.079   43.083   -1.997
GRID            47          77.784   39.525    1.756
GRID            48          77.784   39.525   -1.756
GRID            49          38.565   40.678    1.742
GRID            50          38.565   40.678   -1.742
GRID            51          46.908   37.267    2.082
GRID            52          46.908   37.267   -2.082
GRID            53          55.555   33.732    2.438
GRID            54          55.555   33.732   -2.438
GRID            55          64.523   30.067    2.187
GRID            56          64.523   30.067   -2.187
GRID            57          73.830   26.262    1.922
GRID            58          73.830   26.262   -1.922
GRID            59          32.331   28.347    1.896
GRID            60          32.331   28.347   -1.896
GRID            61          41.214   24.716    2.265
GRID            62          41.214   24.716   -2.265
GRID            63          50.420   20.953    2.651
GRID            64          50.420   20.953   -2.651
GRID            65          59.967   17.050    2.376
GRID            66          59.967   17.050   -2.376
GRID            67          69.876   13.000    2.088
GRID            68          69.876   13.000   -2.088
GRID            69          25.166   14.173    2.073
GRID            70          25.166   14.173   -2.073
GRID            71          35.583   12.304    2.446
GRID            72          35.583   12.304   -2.446
GRID            73          46.181   10.403    2.827
GRID            74          46.181   10.403   -2.827
GRID            75          56.964    8.469    2.502
GRID            76          56.964    8.469   -2.502
GRID            77          67.938    6.500    2.169
GRID            78          67.938    6.500   -2.169
GRID            79          18.000    0.000    2.250
GRID            80          18.000    0.000   -2.250
GRID            81          30.000    0.000    2.625
GRID            82          30.000    0.000   -2.625
GRID            83          42.000    0.000    3.000
GRID            84          42.000    0.000   -3.000
GRID            85          54.000    0.000    2.625
GRID            86          54.000    0.000   -2.625
GRID            87          66.000    0.000    2.250
GRID            88          66.000    0.000   -2.250
ENDDATA
```

Appendix C

Sample ASTROS Output



ASSIGN DATABASE ICW RON NEW DELETE

INTERMEDIATE COMPLEXITY WING ASTROS VERSION 2 6/29/88

B O U N D A R Y C O N D I T I O N S U M M A R Y F O R B O U N D A R Y C O N D I T I O N 1

STATICS/NORMAL MODES				DYNAMIC RESPONSE								
					MODAL				DIRECT			
STATICS	MASS	MODES	SAERO	FLUTTER	TRANS	FREQ	BLAST		TRANS	FREQ	BLAST	
YES	NO	NO	NO	NO	NO	NO	NO		NO	NO	NO	

D I S P L A C E M E N T V E C T O R

POINT ID.	TYPE	T1	T2	T3	R1	R2	R3
1	G	-1.53256E-01	-3.38467E-01	1.50727E+01	0.00000E+00	0.00000E+00	0.00000E+00
2	G	1.53256E-01	3.38467E-01	1.50727E+01	0.00000E+00	0.00000E+00	0.00000E+00
3	G	-1.66921E-01	-3.64387E-01	1.60501E+01	0.00000E+00	0.00000E+00	0.00000E+00
4	G	1.66921E-01	3.64387E-01	1.60501E+01	0.00000E+00	0.00000E+00	0.00000E+00
5	G	-2.11564E-01	-4.69416E-01	1.70456E+01	0.00000E+00	0.00000E+00	0.00000E+00
6	G	2.11564E-01	4.69416E-01	1.70456E+01	0.00000E+00	0.00000E+00	0.00000E+00
7	G	-1.83864E-01	-3.95299E-01	1.81036E+01	0.00000E+00	0.00000E+00	0.00000E+00
8	G	1.83864E-01	3.95299E-01	1.81036E+01	0.00000E+00	0.00000E+00	0.00000E+00
9	G	-1.83057E-01	-3.99944E-01	1.92361E+01	0.00000E+00	0.00000E+00	0.00000E+00
10	G	1.83057E-01	3.99944E-01	1.92361E+01	0.00000E+00	0.00000E+00	0.00000E+00
11	G	-1.67748E-01	-3.77049E-01	1.51186E+01	0.00000E+00	0.00000E+00	0.00000E+00
12	G	1.67748E-01	3.77049E-01	1.51186E+01	0.00000E+00	0.00000E+00	0.00000E+00
13	G	-2.06642E-01	-4.73419E-01	1.51485E+01	0.00000E+00	0.00000E+00	0.00000E+00
14	G	2.06642E-01	4.73419E-01	1.51485E+01	0.00000E+00	0.00000E+00	0.00000E+00
15	G	-1.84103E-01	-4.24446E-01	1.51875E+01	0.00000E+00	0.00000E+00	0.00000E+00
16	G	1.84103E-01	4.24446E-01	1.51875E+01	0.00000E+00	0.00000E+00	0.00000E+00
17	G	-1.64595E-01	-3.86744E-01	1.51950E+01	0.00000E+00	0.00000E+00	0.00000E+00
18	G	1.64595E-01	3.86744E-01	1.51950E+01	0.00000E+00	0.00000E+00	0.00000E+00
19	G	-1.49108E-01	-3.21394E-01	1.08286E+01	0.00000E+00	0.00000E+00	0.00000E+00
20	G	1.49108E-01	3.21394E-01	1.08286E+01	0.00000E+00	0.00000E+00	0.00000E+00
21	G	-1.66272E-01	-3.78739E-01	1.08925E+01	0.00000E+00	0.00000E+00	0.00000E+00
22	G	1.66272E-01	3.78739E-01	1.08925E+01	0.00000E+00	0.00000E+00	0.00000E+00
23	G	-1.92659E-01	-4.65074E-01	1.09003E+01	0.00000E+00	0.00000E+00	0.00000E+00
24	G	1.92659E-01	4.65074E-01	1.09003E+01	0.00000E+00	0.00000E+00	0.00000E+00
25	G	-1.57187E-01	-4.11375E-01	1.09049E+01	0.00000E+00	0.00000E+00	0.00000E+00
26	G	1.57187E-01	4.11375E-01	1.09049E+01	0.00000E+00	0.00000E+00	0.00000E+00
27	G	-1.33547E-01	-3.85815E-01	1.08231E+01	0.00000E+00	0.00000E+00	0.00000E+00
28	G	1.33547E-01	3.85815E-01	1.08231E+01	0.00000E+00	0.00000E+00	0.00000E+00
29	G	-1.43891E-01	-2.96582E-01	7.28257E+00	0.00000E+00	0.00000E+00	0.00000E+00
30	G	1.43891E-01	2.96582E-01	7.28257E+00	0.00000E+00	0.00000E+00	0.00000E+00
31	G	-1.51452E-01	-3.44520E-01	7.34516E+00	0.00000E+00	0.00000E+00	0.00000E+00
32	G	1.51452E-01	3.44520E-01	7.34516E+00	0.00000E+00	0.00000E+00	0.00000E+00
33	G	-1.67551E-01	-4.25978E-01	7.32498E+00	0.00000E+00	0.00000E+00	0.00000E+00
34	G	1.67551E-01	4.25978E-01	7.32498E+00	0.00000E+00	0.00000E+00	0.00000E+00
35	G	-1.31116E-01	-3.76161E-01	7.27819E+00	0.00000E+00	0.00000E+00	0.00000E+00
36	G	1.31116E-01	3.76161E-01	7.27819E+00	0.00000E+00	0.00000E+00	0.00000E+00
37	G	-1.08451E-01	-3.56558E-01	7.12604E+00	0.00000E+00	0.00000E+00	0.00000E+00
38	G	1.08451E-01	3.56558E-01	7.12604E+00	0.00000E+00	0.00000E+00	0.00000E+00
39	G	-1.27474E-01	-2.49420E-01	4.42989E+00	0.00000E+00	0.00000E+00	0.00000E+00
40	G	1.27474E-01	2.49420E-01	4.42989E+00	0.00000E+00	0.00000E+00	0.00000E+00
41	G	-1.30253E-01	-2.91875E-01	4.49267E+00	0.00000E+00	0.00000E+00	0.00000E+00
42	G	1.30253E-01	2.91875E-01	4.49267E+00	0.00000E+00	0.00000E+00	0.00000E+00
43	G	-1.39237E-01	-3.65373E-01	4.46200E+00	0.00000E+00	0.00000E+00	0.00000E+00
44	G	1.39237E-01	3.65373E-01	4.46200E+00	0.00000E+00	0.00000E+00	0.00000E+00
45	G	-9.86454E-02	-3.17122E-01	4.38459E+00	0.00000E+00	0.00000E+00	0.00000E+00
46	G	9.86454E-02	3.17122E-01	4.38459E+00	0.00000E+00	0.00000E+00	0.00000E+00
47	G	-7.50115E-02	-3.04456E-01	4.17675E+00	0.00000E+00	0.00000E+00	0.00000E+00
48	G	7.50115E-02	3.04456E-01	4.17675E+00	0.00000E+00	0.00000E+00	0.00000E+00
49	G	-1.05525E-01	-1.96152E-01	2.32284E+00	0.00000E+00	0.00000E+00	0.00000E+00
50	G	1.05525E-01	1.96152E-01	2.32284E+00	0.00000E+00	0.00000E+00	0.00000E+00
51	G	-1.03731E-01	-2.19895E-01	2.37364E+00	0.00000E+00	0.00000E+00	0.00000E+00
52	G	1.03731E-01	2.19895E-01	2.37364E+00	0.00000E+00	0.00000E+00	0.00000E+00
53	G	-1.03906E-01	-2.76597E-01	2.33001E+00	0.00000E+00	0.00000E+00	0.00000E+00
54	G	1.03906E-01	2.76597E-01	2.33001E+00	0.00000E+00	0.00000E+00	0.00000E+00
55	G	-6.23808E-02	-2.32708E-01	2.24007E+00	0.00000E+00	0.00000E+00	0.00000E+00
56	G	6.23808E-02	2.32708E-01	2.24007E+00	0.00000E+00	0.00000E+00	0.00000E+00
57	G	-4.13869E-02	-2.32492E-01	2.00472E+00	0.00000E+00	0.00000E+00	0.00000E+00
58	G	4.13869E-02	2.32492E-01	2.00472E+00	0.00000E+00	0.00000E+00	0.00000E+00
59	G	-7.97066E-02	-1.15101E-01	9.39395E-01	0.00000E+00	0.00000E+00	0.00000E+00
60	G	7.97066E-02	1.15101E-01	9.39395E-01	0.00000E+00	0.00000E+00	0.00000E+00
61	G	-7.47265E-02	-1.39241E-01	9.70518E-01	0.00000E+00	0.00000E+00	0.00000E+00
62	G	7.47265E-02	1.39241E-01	9.70518E-01	0.00000E+00	0.00000E+00	0.00000E+00
63	G	-7.06392E-02	-1.74268E-01	9.25347E-01	0.00000E+00	0.00000E+00	0.00000E+00
64	G	7.06392E-02	1.74268E-01	9.25347E-01	0.00000E+00	0.00000E+00	0.00000E+00
65	G	-3.28077E-02	-1.32668E-01	8.33358E-01	0.00000E+00	0.00000E+00	0.00000E+00
66	G	3.28077E-02	1.32668E-01	8.33358E-01	0.00000E+00	0.00000E+00	0.00000E+00
67	G	-1.42193E-02	-1.42721E-01	5.98004E-01	0.00000E+00	0.00000E+00	0.00000E+00
68	G	1.42193E-02	1.42721E-01	5.98004E-01	0.00000E+00	0.00000E+00	0.00000E+00
69	G	-4.11256E-02	-4.11276E-02	1.50943E-01	0.00000E+00	0.00000E+00	0.00000E+00
70	G	4.11256E-02	4.11275E-02	1.50943E-01	0.00000E+00	0.00000E+00	0.00000E+00
71		-4.48916E-02	-5.97118E-02	2.18081E-01	0.00000E+00	0.00000E+00	0.00000E+00
72	G	4.48916E-02	5.97118E-02	2.18081E-01	0.00000E+00	0.00000E+00	0.00000E+00
73	G	-4.46456E-02	-8.83522E-02	2.58358E-01	0.00000E+00	0.00000E+00	0.00000E+00
74	G	4.46456E-02	8.83522E-02	2.58358E-01	0.00000E+00	0.00000E+00	0.00000E+00
75	G	-1.93293E-02	-5.85603E-02	2.81571E-01	0.00000E+00	0.00000E+00	0.00000E+00
76	G	1.93293E-02	5.85603E-02	2.81571E-01	0.00000E+00	0.00000E+00	0.00000E+00
77	G	-5.26855E-03	-8.07794E-02	1.87589E-01	0.00000E+00	0.00000E+00	0.00000E+00
78	G	5.26855E-03	8.07794E-02	1.87589E-01	0.00000E+00	0.00000E+00	0.00000E+00
79	G	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
80	G	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
81	G	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
82	G	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
83	G	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
84	G	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
85	G	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
86	G	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
87	G	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
88	G	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00

STATICS ANALYSIS: BOUNDARY 1, SUBCASE 1

STRESSES IN QUADRILATERAL MEMBRANES (QDMEM1)

ELEMENT ID	LAYER NO	STRESSES IN ELEMENT COORD SYSTEM NORMAL-X	NORMAL-Y	SHEAR-XY	PRINCIPAL STRESS ANGLE	PRINCIPAL STRESSES MAJOR	MINOR	MAX SHEAR
3	0	-2.493564E+04	-5.225000E+03	2.612795E+04	65.3331	1.284453E+04	-4.300517E+04	2.792485E+04
4	0	2.493564E+04	5.225000E+03	-2.612795E+04	-34.6669	4.300517E+04	-1.284453E+04	2.792485E+04
5	0	-1.391456E+04	-3.417525E+04	8.356375E+03	19.7594	-1.091278E+04	-3.717704E+04	1.313213E+04
6	0	1.391456E+04	3.417525E+04	-8.356375E+03	-70.2406	3.717704E+04	1.091278E+04	1.313213E+04
7	0	-3.438406E+04	-2.278972E+04	3.294600E+04	49.9898	4.865258E+03	-6.203904E+04	3.345215E+04
8	0	3.438406E+04	2.278972E+04	-3.294600E+04	-40.0102	6.203904E+04	-4.865258E+03	3.345215E+04
9	0	-7.008719E+03	-4.979883E+04	-1.288219E+04	-15.5263	-3.429805E+03	-5.337774E+04	2.497397E+04
10	0	7.008719E+03	4.979883E+04	1.288219E+04	74.4737	5.337774E+04	3.429805E+03	2.497397E+04
11	0	-8.915969E+03	-5.322420E+04	1.304433E+04	15.2448	-5.360955E+03	-5.677922E+04	2.570913E+04
12	0	8.915969E+03	5.322420E+04	-1.304433E+04	-74.7552	5.677922E+04	5.360955E+03	2.570913E+04
13	0	-9.701406E+03	-6.370888E+04	1.399875E+03	1.4838	-9.665145E+03	-6.374514E+04	2.704000E+04
14	0	9.701406E+03	6.370888E+04	-1.399875E+03	-88.5162	6.374514E+04	9.665145E+03	2.704000E+04
15	0	-8.123563E+03	-5.948128E+04	1.630808E+04	15.7182	-1.533988E+03	-6.407086E+04	3.126843E+04
16	0	8.123563E+03	5.948128E+04	-1.630808E+04	-74.2818	6.407086E+04	1.533988E+03	3.126843E+04
17	0	-2.897320E+03	-5.881309E+04	-7.810977E+03	-7.8047	-1.826689E+03	-5.988373E+04	2.902852E+04
18	0	2.897320E+03	5.881309E+04	7.810977E+03	82.1953	5.988373E+04	1.826689E+03	2.902852E+04
19	0	-5.921461E+03	-7.214231E+04	8.153492E+03	6.9170	-4.932328E+03	-7.313145E+04	3.409956E+04
20	0	5.921461E+03	7.214231E+04	-8.153492E+03	-83.0830	7.313145E+04	4.932328E+03	3.409956E+04
21	0	-4.059875E+03	-7.454122E+04	-3.055547E+03	-2.4777	-3.927660E+03	-7.467344E+04	3.537289E+04
22	0	4.059875E+03	7.454122E+04	3.055547E+03	87.5223	7.467344E+04	3.927660E+03	3.537289E+04
23	0	-1.033609E+03	-6.862981E+04	1.379294E+04	11.1001	1.672492E+03	-7.133591E+04	3.650420E+04
24	0	1.033609E+03	6.862981E+04	-1.379294E+04	-78.8999	7.133591E+04	-1.672492E+03	3.650420E+04
25	0	-1.935864E+03	-6.905132E+04	-8.192309E+03	-6.8595	-9.501602E+02	-7.003683E+04	3.454334E+04
26	0	1.935864E+03	6.905132E+04	8.192309E+03	83.1405	7.003683E+04	9.501602E+02	3.454334E+04
27	0	-2.925473E+03	-7.907542E+04	6.127383E+03	4.5711	-2.435586E+03	-7.956531E+04	3.856486E+04
28	0	2.925473E+03	7.907542E+04	-6.127383E+03	-85.4289	7.956531E+04	2.435586E+03	3.856486E+04
29	0	-2.729094E+03	-8.336064E+04	-5.894639E+03	-4.1598	-2.300402E+03	-8.377934E+04	4.073947E+04
30	0	2.729094E+03	8.336064E+04	5.894639E+03	85.8404	8.377934E+04	2.300402E+03	4.073947E+04
31	0	-1.014906E+03	-7.969414E+04	1.192598E+04	8.4325	7.530703E+02	-8.146212E+04	4.110759E+04
32	0	1.014906E+03	7.969414E+04	-1.192598E+04	-81.5676	8.146212E+04	-7.530703E+02	4.110759E+04
33	0	-3.628391E+03	-7.655554E+04	-6.707250E+03	-5.2113	-3.016845E+03	-7.716729E+04	3.707532E+04
34	0	3.628391E+03	7.655554E+04	6.707250E+03	84.7887	7.716729E+04	3.016845E+03	3.707532E+04
35	0	-5.054242E+03	-9.033052E+04	3.691273E+03	2.4739	-4.894762E+03	-9.049001E+04	4.279763E+04
36	0	5.054242E+03	9.033052E+04	-3.691273E+03	-87.5261	9.049001E+04	4.894762E+03	4.279763E+04
37	0	-3.261508E+03	-9.552055E+04	-9.832328E+03	-6.0162	-2.225285E+03	-9.655677E+04	4.716575E+04
38	0	3.261508E+03	9.552055E+04	9.832328E+03	83.9838	9.655677E+04	2.225285E+03	4.716575E+04
39	0	-6.848320E+02	-8.844767E+04	9.074285E+03	5.8418	2.435859E+02	-8.937609E+04	4.480984E+04
40	0	6.848320E+02	8.844767E+04	-9.074285E+03	-84.1582	8.937609E+04	-2.435859E+02	4.480984E+04
41	0	-5.323932E+03	-7.664077E+04	-4.730389E+03	-3.7783	-5.011535E+03	-7.695316E+04	3.597082E+04
42	0	5.323932E+03	7.664077E+04	4.730389E+03	86.2217	7.695316E+04	5.011535E+03	3.597082E+04
43	0	-8.316775E+03	-9.136388E+04	-7.966802E+01	-0.0550	-8.316895E+03	-9.136396E+04	4.152363E+04
44	0	8.316775E+03	9.136388E+04	7.966802E+01	89.9450	9.136396E+04	8.316895E+03	4.152363E+04
45	0	-4.868583E+03	-9.669329E+04	-1.475188E+04	-8.9062	-2.556832E+03	-9.900502E+04	4.822410E+04
46	0	4.868583E+03	9.669329E+04	1.475188E+04	81.0938	9.900502E+04	2.556832E+03	4.822410E+04
47	0	-4.751855E+02	-9.074513E+04	4.815422E+03	3.0449	-2.190352E+02	-9.100128E+04	4.539113E+04
48	0	4.751855E+02	9.074513E+04	-4.815422E+03	-86.9551	9.100128E+04	2.190352E+02	4.539113E+04
49	0	-1.215178E+04	-7.107522E+04	-1.969471E+03	-1.9122	-1.208603E+04	-7.114098E+04	2.952748E+04
50	0	1.215178E+04	7.107522E+04	1.969471E+03	88.0878	7.114098E+04	1.208603E+04	2.952748E+04
51	0	-1.477445E+04	-8.703127E+04	-2.879759E+03	-2.2787	-1.465986E+04	-8.714586E+04	3.624300E+04
52	0	1.477445E+04	8.703127E+04	2.879759E+03	87.7213	8.714586E+04	1.465986E+04	3.624300E+04
53	0	-1.153305E+04	-9.542927E+04	-1.999406E+04	-12.7421	-7.011750E+03	-9.995058E+04	4.646941E+04
54	0	1.153305E+04	9.542927E+04	1.999406E+04	77.2579	9.995058E+04	7.011750E+03	4.646941E+04
55	0	-5.822393E+03	-1.007822E+05	-3.758769E+03	-2.2632	-5.673840E+03	-1.009308E+05	4.762846E+04
56	0	5.822393E+03	1.007822E+05	3.758769E+03	87.7368	1.009308E+05	5.673840E+03	4.762846E+04
57	0	-1.162301E+04	-4.821862E+04	1.020169E+04	14.5706	-8.971250E+03	-5.007038E+04	2.094956E+04
58	0	1.162301E+04	4.821862E+04	-1.020169E+04	-75.4294	5.007038E+04	8.971250E+03	2.094956E+04
59	0	-1.921732E+04	-8.103973E+04	1.063459E+04	9.4926	-1.743912E+04	-8.281793E+04	3.268941E+04
60	0	1.921732E+04	8.103973E+04	-1.063459E+04	-80.5074	8.281793E+04	1.743912E+04	3.268941E+04
61	0	-1.644744E+04	-9.358179E+04	2.169073E+03	1.6095	-1.638649E+04	-9.364274E+04	3.862813E+04
62	0	1.644744E+04	9.358179E+04	-2.169073E+03	-88.3905	9.364274E+04	1.638649E+04	3.862813E+04
63	0	-2.210669E+04	-1.082581E+05	3.196876E+03	2.1222	-2.198822E+04	-1.083766E+05	4.319418E+04
64	0	2.210669E+04	1.082581E+05	-3.196876E+03	-87.8778	1.083766E+05	2.198822E+04	4.319418E+04

STATICS ANALYSIS: BOUNDARY 1, SUBCASE 1

S T R E S S E S I N T R I A N G U L A R M E M B R A N E S (T R M E M)

ELEMENT ID.	LAYER NO.	STRESSES IN ELEMENT COORD SYSTEM NORMAL-X	NORMAL-Y	SHEAR-XY	PRINCIPAL STRESS ANGLE	PRINCIPAL STRESSES MAJOR	MINOR	MAX SHEAR
1	0	1.323869E+04	-1.020675E+04	-8.416094E+03	-17.8379	1.594693E+04	-1.291500E+04	1.443097E+04
2	0	-1.323869E+04	1.020675E+04	8.416094E+03	72.1622	1.291500E+04	-1.594693E+04	1.443097E+04

STATICS ANALYSIS: BOUNDARY 1, SUBCASE 1

S T R E S S E S I N S H E A R P A N E L S (S H E A R)

ELEMENT ID.	MAX SHEAR	AVERAGE SHEAR	SAFETY MARGIN	ELEMENT ID.	MAX SHEAR	AVERAGE SHEAR	SAFETY MARGIN
65	9.210000E+03	-7.985689E+03	1.0E+30	66	5.524000E+03	-4.878268E+03	1.0E+30
67	1.710892E+04	-1.510896E+04	1.0E+30	68	1.054156E+04	-1.607678E+04	1.0E+30
69	1.834200E+04	1.554920E+04	1.0E+30	70	1.963200E+04	1.691753E+04	1.0E+30
71	8.348507E+03	7.653504E+03	1.0E+30	72	2.237213E+04	1.989332E+04	1.0E+30
73	4.066000E+03	3.449971E+03	1.0E+30	74	1.352300E+04	1.166491E+04	1.0E+30
75	1.287350E+04	-1.163700E+04	1.0E+30	76	2.714265E+03	-2.409632E+03	1.0E+30
77	8.721500E+03	7.405316E+03	1.0E+30	78	1.645800E+04	1.420787E+04	1.0E+30
79	6.788029E+03	-6.131640E+03	1.0E+30	80	4.716211E+03	4.183856E+03	1.0E+30
81	1.080700E+04	9.181279E+03	1.0E+30	82	1.749625E+04	1.511384E+04	1.0E+30
83	2.125704E+03	-1.919040E+03	1.0E+30	84	1.042047E+04	9.238796E+03	1.0E+30
85	2.330800E+04	1.981249E+04	1.0E+30	86	2.612588E+04	2.258947E+04	1.0E+30
87	4.084244E+03	3.685403E+03	1.0E+30	88	1.851117E+04	1.640405E+04	1.0E+30
89	3.848934E+04	3.272966E+04	1.0E+30	90	3.676302E+04	3.179983E+04	1.0E+30
91	2.189809E+04	1.974433E+04	1.0E+30	92	3.857693E+04	3.418434E+04	1.0E+30
93	4.024393E+04	3.457490E+04	1.0E+30	94	3.738680E+04	3.268766E+04	1.0E+30
95	2.203178E+04	1.964453E+04	1.0E+30	96	3.885518E+04	3.402794E+04	1.0E+30
97	2.501600E+04	2.218526E+04	1.0E+30	98	2.065000E+04	1.855005E+04	1.0E+30
99	2.048700E+04	1.859543E+04	1.0E+30	100	1.843281E+04	1.686566E+04	1.0E+30
101	1.429575E+04	1.318175E+04	1.0E+30	102	3.447203E+03	3.185435E+03	1.0E+30
103	1.436901E+04	-1.328311E+04	1.0E+30	104	3.342700E+04	3.166358E+04	1.0E+30
105	3.299100E+04	2.931614E+04	1.0E+30	106	3.337650E+04	3.003013E+04	1.0E+30
107	3.385625E+04	3.077031E+04	1.0E+30	108	3.335009E+04	3.056369E+04	1.0E+30
109	3.101364E+04	2.862189E+04	1.0E+30	110	2.726970E+04	2.562482E+04	1.0E+30
111	2.621402E+04	2.474693E+04	1.0E+30	112	2.011300E+04	1.808622E+04	1.0E+30
113	3.761600E+04	3.342084E+04	1.0E+30	114	3.674075E+04	3.310516E+04	1.0E+30
115	3.626563E+04	3.299937E+04	1.0E+30	116	3.533713E+04	3.241692E+04	1.0E+30
117	3.517681E+04	3.249136E+04	1.0E+30	118	3.728138E+04	3.591512E+04	1.0E+30
119	4.104378E+04	3.959278E+04	1.0E+30				

STATICS ANALYSIS: BOUNDARY 1, SUBCASE 1

S T R E S S E S I N R O D E L E M E N T S (R O D)

ELEMENT ID.	AXIAL STRESS	SAFETY MARGIN	TORSIONAL STRESS	SAFETY MARGIN	ELEMENT ID.	AXIAL STRESS	SAFETY MARGIN	TORSIONAL STRESS	SAFETY MARGIN
120	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30	121	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30
122	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30	123	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30
124	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30	125	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30
126	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30	127	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30
128	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30	129	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30
130	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30	131	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30
132	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30	133	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30
134	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30	135	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30
136	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30	137	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30
138	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30	139	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30
140	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30	141	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30
142	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30	143	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30
144	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30	145	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30
146	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30	147	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30
148	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30	149	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30
150	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30	151	-1.250000E-01	1.0E+30	0.000000E+00	1.0E+30
152	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30	153	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30
154	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30	155	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30
156	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30	157	0.000000E+00	1.0E+30	0.000000E+00	1.0E+30
158	-6.250000E-02	1.0E+30	0.000000E+00	1.0E+30					

Appendix D

Random Element Selection

The random element to be damaged was selected thusly: A uniform random number generator was called to provide a number between 0 and 100. The seed number was incremented by one every time the batch file was run. If the number was between 0 and 90, a quad element subroutine was called. If between 90 and 99, a shear panel subroutine was called. If between 99 and 100, the triangular subroutine was called.

The quad subroutine called another random number, normalized to 0 to 31, corresponding to the number of even quad elements. This number was multiplied by two, to give an even number. Then, that number had 2 added to it to compensate for the first two numbered elements not being quad elements. That number and that number minus 1, were considered the choice of elements, giving an even numbered quad element and the odd numbered element above it. These numbers were returned to the section of the program which determined if they were in the section of the wing eligible for damage.

The shear panel subroutine called a random number, normalized to 0 to 54, corresponding to the number of shear panels. This number was returned to the section of the program which determined if it was in a section eligible for damage.

The triangular subroutine merely jumped to the section of the program which checked to see if the tip was the eligible section.

Appendix E

Weight Estimation Calculations

The original weight was estimated by assuming structural material possessed a unit weight. Thus, the cubic volume of the structure could be used for weight calculations.

Most quad elements were about the same size, and approximately rectangular in shape. Element 29, from the middle of the wing, was selected as representative. Using its nodal coordinates, its side lengths were computed to be about 8.4, 13.8, 9.1, and 13.8 units long. This gave the representative element an area of about 120 square units. Then, the thicknesses of all quad panels were summed up to a single thickness and multiplied by the representative area to give about 480 cubic units.

Next, the volume of the shear panels was estimated. Since the wing had a regular shape, root, mid and tip shear panels could be averaged together to give a good representative panel length. There were two basic panel lengths. Long panel ran from the root to the tip; short panels from the leading to trailing edge. Average long panel length was 14 units; average short panels were 9 units long. Average panel height was 4 units. The thicknesses of all the short panels wre summed, then multiplied by 9 by 4 to give about 24 cubic units. Similarly, the long panels summed up to give an additional 70 cubic units. Rod elements contributed, due to their slenderness, no significant volume (0.05 c.u.) In total, the estimated volume of the wing's structural elements was about 575 cubic units.

Then, using the representative quad panel for area, all the strengthening thicknesses were added up, minus the lightening thicknesses (see Figure 22), to give an additional volume of about 24 c.u. Thus the increase in volume (and weight) was about 4%.

Bibliography

1. Abromowitz, Milton and Irene A. Stegun. Handbook of Mathematical Formulas. New York: Dover Publications, 1972.

2. American Institute of Steel Construction. Manual of Steel Construction. New York, 1970.

3. Ang, Alfredo H-S. and Wilson H. Tang. Probability Concepts in Engineering Planning and Design (Volume II). New York: John Wiley and Sons, 1984.

4. Golden Software, Inc. SURFER. Golden, CO, 1986

5. Hamlin, Talbot. Architecture through the Ages. New York: G. P. Putnam's Sons, 1953.

6. Hinrichsen, Maj Ronald, Chief, Structures Branch. Telephone interview. Air Force Wright Aeronautical Laboratories, Wright-Patterson AFB, OH, 15 September 1988.

7. International Conference of Building Officials. Uniform Building Code. Whittier, CA, 1982.

8. Press, William H. Numerical Recipes; The Art of Scientific Computing. New York: Cambridge University Press, 1986.

9. Robinson, Maj David, Assistant Professor, Air Force Institute of Technology. Personal Interview. Wright-Patterson AFB, OH, 9 September 1988.

10. Spotts, M. F. Design of Machine Elements (Third Edition). Englewood Cliffs, NJ: Prentice-Hall, Inc., 1961.

VITA

Captain Paul R. Bryant [redacted] He enlisted in the Navy in 1972, serving as a Sonar Technician aboard submarines and destroyers for six years. He served four years in the Naval Reserve while pursuing his Bachelor of Science degree in Civil Engineering at California State University, Fresno, graduating Magnu Cum Laude in 1983. He received his commission upon graduation from Officer Training School in March of 1984. His initial assignment was to the 831 Civil Engineering Squadron at George Air Force Base, California, where he served as design engineer and as head of the programming branch. He entered the School of Engineering, Air Force Institute of Technology, in June, 1987.

[redacted]

unclassified
SECURITY CLASSIFICATION OF THIS PAGE

ADA206135

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION [redacted]lassified	1b. RESTRICTIVE MARKINGS none
[redacted]ECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/GA/AA/89M-01	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION School of Engineering 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION AFIT/ENY
6c. ADDRESS (City, State, and ZIP Code) Air Force Institute of Technology Wright-Patterson AFB, OH 45433-6583	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)

A DEMONSTRATION OF THE METHOD OF STOCHASTIC FINITE ELEMENT ANALYSIS

12. PERSONAL AUTHOR(S)
[redacted]l R. Bryant, Captain, USAF

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
MS Thesis	FROM _____ TO _____	1989 March	96

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	-Finite Element Analysis -Stochastic Processes
12	01		

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Major David G. Robinson, Advisor

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
[redacted] UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL [redacted]avid G. Robinson, Assistant Professor	22b. TELEPHONE (Include Area Code) (513) 255-2362 / 22c. OFFICE SYMBOL AFIT/ENY

DD Form 1473, JUN 86 *Previous editions are obsolete.*

SECURITY CLASSIFICATION OF THIS PAGE
unclassified

Abstract

Finite element analysis has been used as a design tool for many years, with structural reliability being ensured through use of a liberal factor of safety. Unfortunately, the safety factor is a blanket insurance against all hazards, and a designer has no way to optimize a structure against any particular hazard. This is particularly troublesome in the fields of aero/astro design, where every bit of mass must serve to maximum utility.

The method of Stochastic Finite Element Analysis allows a designer to model any loading or hazard condition as closely to reality as desired by using an appropriate probability distribution function. Through a Monte Carlo simulation, the finite element model is subjected to the probability functions. The cumulative output is analyzed for trends in failure probability and the design is altered to enhance its reliability, repeating the process until the desired level of reliability is achieved. The resulting design is optimal for the imposed conditions, and compared to a structure designed with a traditional factor of safety approach, is either lighter or more reliable. This demonstration revealed that for similar reliabilities, a stochastically designed wing was 20% lighter than a wing strengthened by the factor of safety.

The major drawback in applying the method of stochastic finite element analysis is that very large, complex models can require extraordinary amounts of computer resources.